

APPROVED:

Gary A. Pope
H. Sychamoori
Larry W. Lake

THE APPLICATION OF IMPROVED NUMERICAL TECHNIQUES
TO 1-D MICELLAR/POLYMER FLOODING SIMULATION

BY

TAKAMASA OHNO, B.E. Pet.E.

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of
MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

August 1981

ACKNOWLEDGEMENT

I would like to express my sincere appreciation and gratitude to Dr. Gary A. Pope for his supervision and encouragement throughout this study.

I am also deeply grateful to Dr. K. Sepehrnoori for valuable assistance and guidance in the numerical analysis and the solution of differential equations.

Thanks and appreciation are also extended to Dr. L. W. Lake for his helpful comments and suggestions.

I would also like to thank the Japan Petroleum Exploration Corporation (JAPEx) for their financial support during this research. This research was also supported by grants from the Department of Energy and the following companies: Core Laboratories; Exxon U.S.A.; INTERCOMP Resources and Engineering; Marathon Oil; Mobil Oil; Shell Development; Tenneco Oil; Cities Service.

Special appreciation is expressed to my wife, Mihoko, for her encouragement and understanding which made this work possible.

T. Ohno

The University of Texas
Austin, Texas

June, 1981

ABSTRACT

Three examples of three phase flow models which have been developed are compared under various conditions. Although the difference in oil recovery and surfactant trapping among the models was rather large with constant salinity, a salinity gradient produced high oil recovery and low surfactant trapping with all three models. Since surfactant trapping is important and it is highly uncertain, this is another reason for designing a micellar flood with a salinity gradient, or something equivalent to a salinity gradient.

The semi-discrete method was applied to a 1-D micellar/polymer flooding simulator. By using a semi-discrete method, the time step size can be controlled and varied to be as large as possible without sacrificing accuracy. The stability limit can also be detected with this method. The method is tested and compared with the fully discrete method in various conditions such as different phase behavior environments and with or without adsorption. In the application of the semi-discrete method, four different ODE integrators were used. Two of them are explicit methods while the other two are implicit methods. Although the implicit methods did not work as well as the explicit methods, there may be some improvement possible. With respect to the computation time, one of the explicit methods which is based on the Runge-Kutta approximation worked best. Although the method can save 20 to 30% computation

time under some conditions, compared with the fully-discrete method, the results are highly problem-dependent. To improve the computation time, two methods are suggested. One is to check the error only in the oil or water component rather than all components or any other one component such as surfactant. The other is to check absolute error instead of relative error and multiply by a small conservative factor to the calculated time step size.

The stability was analyzed for the oil bank, and for the surfactant front. The former imposes a rather constant limitation on the time step size continuously until the plateau of the oil bank is completely produced. Although approximate, the stability analysis for the surfactant front suggests an unconditional local instability, which is caused by the change in the fractional flow curve due to the surfactant.

TABLE OF CONTENTS

Acknowledgement	iii
Abstract	iv
Table of Contents	vi
List of Tables	ix
List of Figures	xi
 Chapter	 Page
I. INTRODUCTION	1
II. LITERATURE SURVEY AND REVIEW OF PREVIOUS WORK	7
III. DESCRIPTION OF MICELLAR/POLYMER FLOODING SIMULATOR	10
3.1 Basic Assumptions and Governing Continuity Equations	10
3.2 Auxiliary Functional Relationships	17
3.2.1 Salinity	19
3.2.2 Binodal Curve and Distribution Curve	20
3.2.3 Adsorption	24
3.2.4 Phase Viscosity	25
3.2.5 Interfacial Tension	26
3.2.6 Trapping Function	27
3.2.7 Relative Permeability	30
3.2.8 Other Features	32

3.3	Solution Procedure	32
IV.	THREE PHASE FLOW MODEL	44
4.1	Pope's Model	44
4.2	Hirasaki's Model	45
4.3	Lake's Model	48
4.4	Comparison of Each Model	50
V.	ORDINARY DIFFERENTIAL EQUATION INTEGRATORS	63
5.1	Runge-Kutta-Fehlberg Methods	64
5.2	Multi Step Methods	70
5.3	Stability Region and Stiffness	72
VI.	RESULTS AND DISCUSSIONS	82
6.1	Basic Input Data and Example Results	84
6.2	Comparison of Each Method for Type II(-)	
	Phase Behavior without Adsorption Case	87
6.3	Effect of Adsorption and Phase Behavior	93
6.4	Effect of Each Component on Time Step	
	Size Selection	95
6.5	Additional Test Runs and Summary of RK1(2)	
	and RK1 for Type II(-) Phase Behavior Envir-	
	onment without Adsorption	100
6.6	Stability Requirement before Surfactant Break-	
	through	101
6.7	Analysis of Stability at Blocks Where Sur-	
	factant is Present	109

VII. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE	
STUDIES	171
NOMENCLATURE	176
APPENDICES	
A. PHASE BEHAVIOR CONCEPT	180
B. DERIVATION OF CONTINUITY EQUATIONS FOR MULTIPHASE	
MULTICOMPONENT FLOW	185
C. COMPUTER PROGRAM	192
REFERENCES	233
VITA	239

LIST OF TABLES

<u>Table</u>	<u>Page</u>
3.1 Equations used to calculate phase concentrations	35
4.1 Comparison of oil recovery and surfactant trapped (0.1 PV of 3% surfactant slug was injected, salinity is constant)	52
4.2 Comparison of oil recovery and surfactant trapped (0.1 PV of 6% surfactant slug was injected. Salinity is constant.)	53
4.3 Comparison of oil recovery and surfactant trapped for two different salinity gradient (0.1 PV of 3% surfactant slug was injected)	54
4.4 Input data used to compare three phase flow models	55
5.1 Coefficients for RK1(2)	67
6.1 Basic input data used to test semi-discrete method	114
6.2 Comparison of FDE method and semi-discrete method with various ODE integrators (Type II(-), no adsorption)	117
6.3 Comparison of FDE and RK1 (Type II(-), no adsorption)	118
6.4 Results of predictor-corrector methods at 0.1 PV injected	119

6.5	Comparison of FDE, RK1(2), and RK1 for Type II(-), with adsorption	120
6.6	Comparison of FDE and RK1 for Type II(+), no adsorp- tion	121
6.7	Comparison of FDE, RK1(2), and RK1 for Type II(+), with adsorption	122
6.8	Comparison of FDE and RK1 for Type III	123
6.9	Effect of adsorption and phase behavior environment on the efficiency of RK1 (ERR = 0.01)	124
6.10	Effect of each component on time step size selec- tion (RK1, ERR = 0.01, Type II(-), no adsorption)	125
6.11	Improvement in time step size selection by checking errors for oil component (RK1, ERR = 0.01)	126
6.12	Test runs with YBIAS = 1.0 (RK1(2) and RK1, Type II(-), no adsorption)	127
6.13	Summary of RK1(2) and RK1 (Type II(-), no adsorption)	128
6.14	Aqueous phase profile of Run 207 at 0.5 PV injection (Type II(-), no adsorption)	129
6.15	Legend for the total concentration history plots	130

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
3.1 Method of lines	16
3.2 Definition of phase number	36
3.3 Effect of G parameters on calculated IFT curve	37
3.4 Effect of G parameters on calculated IFT curve	38
3.5 Effect of G parameters on calculated IFT curve	39
3.6 Effect of G parameters on calculated IFT curve	40
3.7 Normalized residual saturation versus capillary number	41
3.8 Interdependence of variables without shear rate effect on polymer solution	42
3.9 Solution procedure in the simulator	43
4.1 Basic idea of phase trapping employed in Hirasaki's model	57
4.2 Phase diagrams and interfacial tensions in Type III phase behavior environment with different salinities	58
4.3 Comparisons of final oil recoveries and surfactant trappings among three different three phase flow models (0.1 PV of 3% surfactant slug is injected)	59
4.4 Comparisons of final oil recoveries and surfactant trappings among three different three phase flow models (0.1 PV of 6% surfactant slug is injected)	60

4.5	Comparison of production histories among three different three phase flow models (constant salinity $C_{SE} = 0.82$)	61
4.6	Comparison of production histories among three different three phase flow models (salinity gradient)	62
5.1	Stability region for RK1(2)	79
5.2	Stability region for Adams' method	80
5.3	Stability region for Gear's methods	81
6.1	Total concentration profile at 0.25 PV for Run 207 (FDE method, $\Delta t_D = 0.001$ PV, Type II(-), no adsorption)	131
6.2	Phase saturation profile at 0.25 PV for Run 207	132
6.3	Fractional flow profile at 0.25 PV for Run 207	133
6.4	Microemulsion phase concentraton profile at 0.25 PV for Run 207	134
6.5	Oleic phase saturation and its residual saturation profile at 0.25 PV for Run 207	135
6.6	Interfacial tension profile at 0.25 PV for Run 207	136
6.7	Total relative mobility profile at 0.25 PV for Run 207	137
6.8	History of total concentration in effluent for Run 207	138
6.9	History of phase cuts and cumulative oil production for Run 207	139

6.10	History of pressure drop (normalized) between producer and injector for Run 207	140
6.11	History of total concentration in effluent for Run 203 (FDE method, $\Delta t_D = 0.002$, Type II(-), no adsorption)	141
6.12	History of total concentration in effluent for Run 204 (FDE method, $\Delta t_D = 0.003$, Type II(-), no adsorption)	142
6.13	History of total concentration in effluent for Run 206 (FDE method, $\Delta t_D = 0.004$, Type II(-), no adsorption)	143
6.14	Histories of total concentration in effluent and time step size for Run 196 (RK1(2), ERR = 0.00005, Type II(-), no adsorption)	144
6.15	Histories of total concentration in effluent and time step size for Run 197 (RK1(2), ERR = 0.0001, Type II(-), no adsorption)	145
6.16	Histories of total concentration in effluent and time step size for Run 198 (RK1(2), ERR = 0.002, Type II(-), no adsorption)	146
6.17	Histories of total concentration in effluent and time step size for Run 498 (Adams' method, ERR = 0.0005, Type II(-), no adsorption)	147

6.18	Histories of total concentration in effluent and time step size for Run 496 (Adams' method, ERR = 0.001, Type II(-), no adsorption)	148
6.19	Histories of total concentration in effluent and time step size for Run 495 (Adams' method, ERR = 0.01, Type II(-), no adsorption)	149
6.20	Histories of total concentration in effluent and time step size for Run 200 (RK1, ERR = 0.01, Type II(-), no adsorption)	150
6.21	Histories of total concentration in effluent and time step size for Run 201 (RK1, ERR = 0.02, Type II(-), no adsorption)	151
6.22	Histories of total concentration in effluent and time step size for Run 202 (RK1, ERR = 0.05, Type II(-), no adsorption)	152
6.23	Histories of total concentration in effluent and time step size for Run 333 (RK1, ERR = 0.01, Type II(-), with adsorption)	153
6.24	Histories of total concentration in effluent and time step size for Run 411 (RK1, ERR = 0.01, Type II(+), no adsorption)	154
6.25	Histories of total concentration in effluent and time step size for Run 395 (RK1, ERR = 0.01, Type II(+), with adsorption)	155

6.26	Histories of total concentration in effluent and time step size for Run 442 (RK1, ERR = 0.01, Type III, no adsorption)	156
6.27	Histories of total concentration in effluent and time step size for Run 463 (RK1, ERR = 0.01, Type III, with adsorption)	157
6.28	Histories of total concentration in effluent and time step size for Run 265 (RK1, ERR = 0.01, Type II(-), no adsorption, no tracer)	158
6.29	Histories of total concentration in effluent and time step size for Run 266 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_1)	159
6.30	Histories of total concentration in effluent and time step size for Run 267 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_2)	160
6.31	Histories of total concentration in effluent and time step size for Run 268 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_3)	161
6.32	Histories of total concentration in effluent and time step size for Run 269 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_4)	162
6.33	Histories of total concentration in effluent and time step size for Run 229 (RK1, ERR = 0.01, waterflooding, no tracer)	163

6.34	Histories of total concentration in effluent and time step size for Run 260 (RK1, ERR = 0.01, miscible displacement with tracer)	164
6.35	Histories of total concentration in effluent and time step size for Run 522 (RK1, ERR = 0.01, Type II(-), with adsorption, error checked only for C_2)	165
6.36	Histories of total concentration in effluent and time step size for Run 446 (RK1, ERR = 0.01, Type III, no adsorption, error checked only for C_2)	166
6.37	Histories of total concentration in effluent and time step size for Run 629 (RK1(2), ERR = 0.0001, PCT = 0.25, YBIAS = 1.0, Type II(-), no adsorption)	167
6.38	Histories of total concentration in effluent and time step size for Run 647 (RK1(2), ERR = 0.0001, PCT = 0.50, YBIAS = 1.0, Type II(-), no adsorption)	168
6.39	Stability region for RK1 or Forward Euler	169
6.40	Explanation of negative value for f'	170

CHAPTER I

INTRODUCTION

Process background

Micellar/polymer flooding has been recognized as one of the most promising enhanced oil recovery techniques as well as CO₂ injection and thermal recovery. Some people refer to micellar flooding as a miscible displacement even though this is not always true. Even if there exist two or three distinct phases associated with the surfactant process, improved oil recovery can be achieved. Three major mechanisms that contribute to improved oil recovery by micellar flooding have been suggested as below

- (1) miscible displacement
- (2) ultra low interfacial tensions¹
- (3) oil swelling or solubilization²

In 1927, Uren and Fahmy³ concluded that the oil recovery obtained by flooding has a definite relationship with interfacial tension between the oil and the displacing fluid. Ever since extensive research has been done, especially in the laboratory, to analyze the mechanisms and efficiency of flooding with surfactants and other chemical agents. The literature⁴ gives a list of representative references and brief summary of the history. In spite of this, however, the optimum method is still under investigation both in the laboratory and in the field.

In designing and optimizing a micellar flooding process, one is confronted with many mechanisms and corresponding physicochemical properties of the rock and fluid interactions that affect performance of a micellar flood. Important properties are phase behavior, interfacial tension, relative permeabilities, viscosities, dispersion, adsorption and cation exchange. Laboratory investigations to analyze process sensitivity to each property are very difficult because they are highly coupled with one another. Accordingly, several numerical simulators for micellar/polymer flooding^{5-16,67,68} have been presented both to aid in the interpretation of laboratory experiments and to scale it up for field applications.

When one tries to simulate micellar flooding numerically, special care must be taken of surfactant transport in porous media. One principal problem is numerical dispersion^{17,18} that may swamp physical dispersion^{19,20}, leading to a front apparently much more smeared than it should be. The numerical dispersion is produced from truncation errors when the spatial and/or time derivatives in a differential equation are approximated as difference quotients. The smeared solution may not cause much trouble if the surfactant slug is injected continuously. However, in actual field operation, the surfactant slug is usually injected as a finite slug, sometimes as low as a few percent of the reservoir pore volume, because of the high cost of chemicals. In such a case, simulated performance may be quite erroneous and lead people to a wrong judgement,

if the numerical dispersion is not treated properly. This is especially true when the phase behavior environment in the reservoir is Type II(+).

The dispersion causes the peak surfactant concentration to decrease, which causes the concentration to fall below the multiphase boundary earlier, resulting in the earlier loss of one mechanism of improved oil recovery: miscibility. Furthermore, in the multiphase region, when the phase behavior environment is Type II(+), decreased surfactant concentration results in greater retardation of the surfactant and loss in the ability to cause oil swelling, another contribution to higher oil recovery.

Another problem associated with the construction of a micellar/polymer simulator is the lack of knowledge about phase trapping and flow character when three phases coexist.

There have been two competing design philosophies for surfactant flooding^{4,7,10} although, as Larson⁷ pointed out, the distinction is a matter of degree. One is to inject a relatively small pore volume (about 3-20%)⁴ of higher concentration surfactant slug, usually with a non-zero oil content. The main mechanism of its oil recovery is miscible displacement: solubilize both oil and water in the reservoir leaving no residual oil since there exists no interfacial tension for single phase flow, until the chemical concentration falls below the multiphase boundary. The other is to inject a large pore volume (about 15-60%) of lower surfactant concentration slug, usually with little oil content. The major mechanism of improved oil recovery is

no longer miscibility in this case. Ultra-low interfacial tension between the aqueous and oleic phases due to the surfactant reduces residual oil saturation and increases oil recovery.

Healy et al.¹ showed experimental results which indicated the lowest interfacial tension between the microemulsion and either the oleic or aqueous phase occurred when three phases coexist. Nelson and Pope² also showed higher efficiency of oil recovery in the Type III phase environment where three phases coexist. Thus, the transport characteristics of three phase flow must be considered and included in the simulator to determine the optimum method of micellar flooding.

Unfortunately, little experimental data which represent three phase flow in a Type III phase environment have been published. So we have to make some hypothetical model based upon reasonable assumptions. A few models for such three phase flow have been presented.^{5,9,37} Although it is very hard to say which is realistic from the simulated results, a comparison is made in Chapter IV among those models just as a reference.

Numerical background

In general, there are two approaches used to solve partial differential equations. One is the fully-discrete model and the other is the semi-discrete method. In a fully-discrete method, time derivatives are discretized and approximated by the finite difference expressions, whereas they are left to be continuous in a semi-discrete method. In both methods, spatial derivatives may be discretized and approximated

as difference quotients: the finite difference methods, or the problem may be formulated as a variational problem in the spatial domain: the finite element methods.²¹⁻²⁵ Furthermore if we consider equations that involve both parabolic and hyperbolic characters, like the well known convection diffusion equation, the finite difference methods can be categorized in two groups; one solves equations as parabolic²⁶⁻³¹ and the other as hyperbolic.³²⁻³⁶

Fully-discrete finite difference methods that solve problems as parabolic are the most common techniques in the area of reservoir simulation, and have been used in micellar/polymer flooding simulators to solve the continuity equations. Those techniques, however, exhibit some inherent problems when one tries to solve the case with small dispersion. When the spatial derivatives are approximated by backward difference expressions, the solution is smeared by numerical dispersion. When the centered difference approximations are used instead, the solution oscillates. To eliminate those problems, one may have to use more grid blocks or nodes which increases computer costs and may be impractical in some cases.

In this research, some semi-discrete methods are applied to a micellar/polymer flooding simulator which uses a parabolic technique of finite difference methods. By using a semi-discrete method, the time step size can be controlled and varied to be as large as possible without sacrificing accuracy. Thus, it can be expected that the semi-discrete method may save computation time.

When a semi-discrete method is applied to solve partial differential equations, they are converted to a system of ordinary differential equations (ODE's), since the time derivatives remain continuous. To solve the resulting ODE's, a Runge-Kutta-Fehlberg (RKF) method, Adams' methods and Gear's backward differentiation methods were first tested. RKF methods are explicit algorithms to integrate with respect to time, whereas the other two are implicit (predictor-corrector) methods, which requires some iterative scheme to solve non-linear equations. The details of these ODE solvers are presented in Chapter V.

After the test of all three ODE solvers, another algorithm which seemed to be more efficient was also examined. This algorithm consists of a combination of first and second order Runge-Kutta approximations. A brief description of the algorithm is given in Chapter VI.

The micellar/polymer flooding simulator used in this research was originally presented by Pope et al.⁵ Then Wang³⁸ and Lin³⁹ made several improvements. The details of the simulator are described in Chapter III.

CHAPTER II

LITERATURE SURVEY AND REVIEW OF PREVIOUS WORK

In most chemical flooding simulators, continuity equations are solved for several components. Although the equations are highly non-linear, their character is quite similar to the well known linear convection diffusion (C-D) equation. Depending on the degree of dispersion, the character of the equations ranges from parabolic to almost hyperbolic.

Among the chemical flooding simulators which have been presented, fully discrete finite difference methods that solve equations as parabolic are the most common techniques. Some authors⁵⁻⁹ employ the analysis of Lantz.¹⁸ Other authors¹¹⁻¹⁴ use higher order accurate approximation.^{28,29} Since these techniques are suitable for parabolic equations, they have inherent problems when the level of dispersion is very low. The Lantz's technique may become impractical because of great computation time, since it requires fine grid spacing to approximate low dispersion. Furthermore the continuity equations are usually solved explicitly, which imposes a strict limitation on the time step size and makes the computation time proportional to the square of number of spatial grid points. The higher order accurate methods, on the other hand, require fewer spatial grids to attain the same level of dispersion. Fine grid spacing, however, is still required for lower dispersion to avoid oscillation. It also involves the problem of small time step size due to explicit solution.

Todd and Chase¹¹ used an automatic time step size control in a chemical flooding simulator. They varied the step size based on the relative changes of variables during the last time step. This technique may be called "the method of relative changes" and distinguished from semi-discrete methods which control time step size based on estimated truncation error made during the last time step.

The method of relative changes is rather widely used in reservoir simulation. Coats⁴⁰ applied the same kind of method to a steam-flood simulator and Grabowski et al.⁴ used its modified form in a general purpose thermal model for in situ combustion and steam. However, these methods only rely on the observation that large changes in the variables mean more error and small changes mean less error. Although the error control is not rigorous, this method can control stability. If the stability condition is not met on the way of continuous integration, the large change in the variables due to instability makes the time step size smaller, and forces it back toward the stability region. However, the point is that a stable scheme does not necessarily mean high accuracy.

Some ordinary differential equation (ODE) solvers have been applied to solve partial differential equations with the semi-discrete method. When an ODE-solver is used for time integration, truncation error made during one time step is estimated and then the time step size is varied according to the estimated error. The semi-discrete method has two advantages over fully discrete methods:

- 1) by changing time step size and the order of integration

scheme, truncation error associated with time integration is kept uniform while being forced to stay within error tolerance which is usually specified by users.

2) time step size is controlled to be as large as possible without sacrificing accuracy.

Sincovec⁴² introduced Gear's all-purpose ODE-solver⁴³ into reservoir simulation problems when he applied the semi-discrete method. Although he had difficulty in solving a highly non-linear problem, he obtained successful results in other problems. Jensen⁴⁴ applied a first order predictor-corrector method, which is based on Gear's approach, to automatically select the time-step size in a finite difference steam injection reservoir simulator. He compared the scheme with the method of relative changes and showed the superiority of his scheme.

Sepehrnoori and Carey⁴⁵ applied several sophisticated ODE-solver programs to stiff and non-stiff initial-value systems arising from representative evolution problems. Basic algorithms used in those programs are Adams' method, Gear's method and the modified (extended stability region) Runge-Kutta method. The efficiencies of those algorithms are compared for each problem. They found that the performance of each method is highly problem dependent.

CHAPTER III

DESCRIPTION OF MICELLAR/POLYMER FLOODING SIMULATOR

3.1 Basic Assumptions and Governing Continuity Equations

The continuity equations for multiphase multicomponent flow are derived based on the following assumptions.

- (1) Isothermal system.
- (2) One-dimensional flow with homogeneous rock properties.
- (3) Rock compressibility is negligible.
- (4) Gravity and capillary pressure are negligible.
- (5) Fluid properties are a function of composition only.
- (6) The volume of a mixture is equal to the sum of individual pure-component volumes: volume does not change upon mixing.
- (7) Pure component densities are constant.
- (8) Local thermodynamic equilibrium exists everywhere.
- (9) Darcy's law applies.
- (10) No chemical reaction occurs (no appearance or disappearance of any species).

Given the above assumptions and some other minor assumptions, the continuity equations for each component i in dimensionless form are

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{\partial (f_j C_{ij})}{\partial x_D} - \sum_{j=1}^M \alpha_{Dj} \frac{\partial}{\partial x_D} \left(f_j \frac{\partial C_{ij}}{\partial x_D} \right) = 0 \quad (3.1)$$

$$i = 1, 2, \dots, \text{NCOMP}$$

where

$$f_j = \frac{k_j/\mu_j}{\sum_{j=1}^M (k_j/\mu_j)}$$

$$t_D = \int_0^t \frac{u_T}{\phi L} dt$$

$$x_D = \frac{x}{L}$$

$$\tilde{C}_i = (1 - \sum_{i=1}^{NCOMP} \bar{C}_i) C_i + \bar{C}_i \quad (3.2)$$

$$C_i = \sum_{j=1}^M S_j C_{ij} \quad (3.3)$$

Definitions (3.2) and (3.3) give

$$\sum_{i=1}^{NCOMP} C_i = \sum_{i=1}^{NCOMP} \tilde{C}_i = 1 \quad (3.4)$$

since

$$\sum_{i=1}^{NCOMP} C_{ij} = \sum_{j=1}^M S_j = 1$$

And

C_i = volume of component i adsorbed per unit pore volume

C_{ij} = concentration of component i in phase j

k_j = effective permeability to phase j

μ_j = viscosity of phase j

u_T = superficial (Darcy) velocity of total phase

ϕ = porosity

L = length of system

x = distance

α_{Dj} = dimensionless longitudinal dispersivity

M and $NCOMP$ are the number of phases and components, respectively.

The derivation of Eq. (3.1) is given in Appendix B.

If assumption (11) below is added, we get

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{\partial (f_j C_{ij})}{\partial x_D} = 0 \quad (3.5)$$

(11) Physical dispersion can be approximated adequately by numerical dispersion by selecting the appropriate grid size and time step.

When equation (3.5) is fully discretized using a backward difference approximation in space and forward difference in time, the actual equations we solve are

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{\partial (f_j C_{ij})}{\partial x_D} - D_{Ni} \approx 0 \quad (3.6)$$

where the last term is called numerical dispersion term and for small

$$D_{Ni} \approx \frac{\Delta x_D}{2} \sum_{j=1}^M \frac{\partial^2 (f_j C_{ij})}{\partial x_D^2} \quad (3.7)$$

More detail of this approximation will be discussed later.

Although assumption (11) is based on the analysis of single-phase flow (linear convection diffusion equation), it may be a reasonable approximation in many cases. Lin³⁹ has tested the numerical difference between this approximation compared to solving equation (3.1) with a very large number of grid blocks to minimize numerical dispersion. He found close agreement, but no way of generalizing this result to other cases has been developed. When assumption (11) is employed, the porous medium is, in effect, being modeled as a series of well-stirred tanks, each of which at each time step dumps a portion of its contents into the next tank forward according to the fractional flow rather than saturation of each phase.

In the simulator, equation (3.1) is solved rather than equation (3.5). However, the former is easily converted to the latter by only setting $\alpha_{Dj} = 0$ in input data. When $\alpha_{Dj} > 0$ it should be noted that the solution obtained includes both physical dispersion and numerical dispersion. In other words, the solution obtained from equation (3.1) always includes more dispersion compared with

the one obtained from equation (3.5) as long as a positive value of α_{Dj} is used. Although negative values of α_{Dj} are non-physical, Lin³⁹ tried some numerical experiments, expecting that numerical dispersion is cancelled somehow by the negative α_{Dj} . The results seem to be successful to some extent. Some oscillation, however, is still inevitable when the desired order of effective dispersion (the sum of numerical and physical dispersion) is low.

Because α_{Dj} was set to be zero for all test runs, the numerical approximation of equation (3.5) rather than (3.1) is discussed here. In other words, assumption (11) remains in this research. The approximation of equation (3.1) is discussed by Lin.³⁹

Equation (3.5) is solved numerically by either the semi-discrete or fully discrete finite difference method. In either case, spatial derivatives are discretized using single-point (one-point) upstream weighting. For the fully discrete method, time derivatives are approximated by forward differences which allow explicit solutions. For semi-discrete methods, time derivatives are continuous, rather than discretized, allowing the application of Ordinary Differential Equation (ODE) solvers (which then contain the time discretization).

Fully discrete finite difference equations are solved explicitly as below

$$(\tilde{C}_i)_{t_D + \Delta t_D} = (\tilde{C}_i)_{t_D} - \frac{\Delta t_D}{\Delta x_D} \sum_{j=1}^M \{ (f_j C_{ij})_{x_D} - (f_j C_{ij})_{x_D - \Delta x_D} \} \quad (3.8)$$

Taking the truncation error into account, the actual equations being solved are

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{(f_j C_{ij})}{\partial x_D} - \frac{1}{2} \Delta x_D \sum_{j=1}^M \frac{\partial^2 (f_j C_{ij})}{\partial x_D^2} + \text{HOT} = 0 \quad (3.9)$$

where HOT means higher order truncation errors. In most cases the time step size Δt_D is forced to be much smaller than Δx_D to keep stability because the explicit method is used. Therefore, the dispersion is controlled by Δx_D , or the number of grid blocks.

When semi-discrete finite difference methods are used, equations (3.10) below are to be solved

$$\left(\frac{d\tilde{C}_i}{dt_D} \right)_{x_D} = - \frac{1}{\Delta x_D} \sum_{j=1}^M \{ (f_j C_{ij})_{x_D} - (f_j C_{ij})_{x_D - \Delta x_D} \} \quad (3.10)$$

where time is retained as a continuous variable. This semi-discrete approach yields an initial value system of ordinary differential equations with respect to time. Because every component in one block affects the right-hand side of equation (3.10) for all components in the same block and the next (downstream) block, all concentrations in all blocks are coupled. Then the number of equations involved in the system is given by the product of the number of components and the number of spatial grid points. For example, when six components and forty points are used, a system of two hundred and forty equations has to be solved. Reordering of each component in each block is done as follows:

$$y_m = (\tilde{C}_i)_k \quad (3.11)$$

where

y_m = reordered variable

$m = (k - 1) * NCOMP + i$

k = block number

i = component number

$NCOMP$ = number of components

This semi-discrete system is solved making use of an ODE solver. Such a technique is sometimes referred to as the method of lines. This name came from the fact that the dependent variable is integrated along the lines of fixed spatial points with varied time as shown in Figure 3.1.

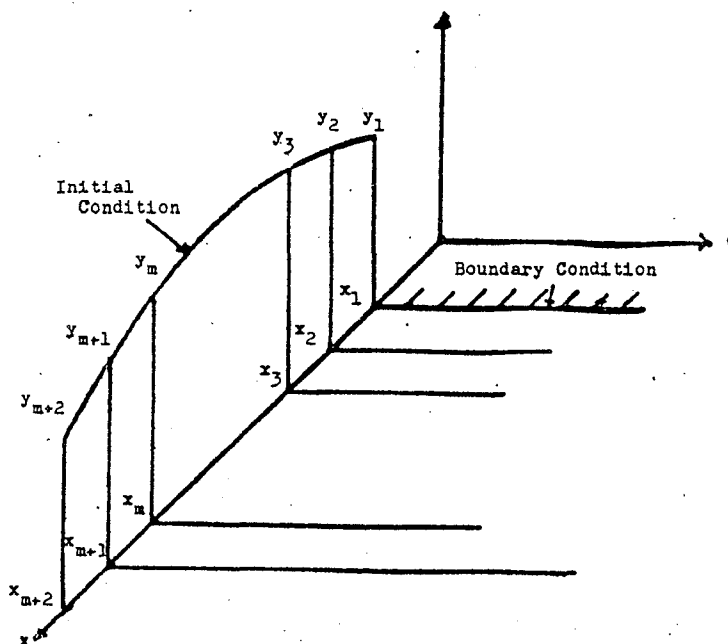


Figure 3.1 Method of lines.

Even though the time derivatives remain continuous, their integration must be done numerically, which means that truncation error associated with the integration are inevitable. However, the degree of such truncation errors can be much smaller than the one produced from fully discrete methods, if compared with the same time step size. Thus the truncation errors, TE, associated with semi-discrete methods come mainly from space discretization

$$TE \cong - \frac{\Delta x_D}{2} \sum_{j=1}^M \frac{\partial^2 (f_j C_{1j})}{\partial x_D^2} \quad (3.12)$$

Even if a semi-discrete method is used, physical dispersion can be approximated by numerical dispersion in a similar way as with the fully discrete method.

Since the derivatives involved in equation (3.5) are first order with respect to both time and space, one of each temporal and spatial boundary conditions are required. Temporal boundary condition (initial condition) has to be given to the simulator by the user to start computation. Usually the initial condition is set to be the post-waterflooding condition. The spatial boundary condition is taken to be the inflow concentrations during each time step (corresponding to the injection of slug or drive).

3.2 Auxiliary functional relationships

In order to solve the continuity equation (3.1), many functional relationships as well as additional assumptions are needed

to obtain C_{ij} and f_j .

Component number and phase number are determined as follows.

At most seven components are considered: (1) water, (2) oil, (3) surfactant or surfactant and cosurfactant, (4) polymer, (5) total anions, (6) calcium ion, and (7) alcohol. The alcohol can be combined with the surfactant as component three. Adding the surfactant and alcohol components together is an approximation. The accuracy depends greatly on the particular system and conditions involved. The maximum number of mobile phases considered is three: (1) aqueous, (2) oleic, and (3) microemulsion. The last one is defined simply as the phase containing the highest concentration of surfactant. The details are shown in Figure 3.2. It should be noted that the number of phases changes from time to time and place to place, depending on the total composition (including salinity), with some phase appearing and some phase disappearing.

The polymer and electrolytes are assumed to occupy negligible volume. The adsorption of water, oil, and alcohol is zero. Polymer (C_4) and calcium (C_6) do adsorb, but occupy no volume. Thus, equation (3.2) is rewritten

$$\tilde{C}_i = (1 - \bar{C}_3)C_i + \bar{C}_i \quad (3.13)$$

The polymer is assumed to be entirely in the most water-rich phase, whereas the electrolytes are assumed to be uniformly distributed in the water component.

3.2.1 Effective salinity

Since the physical properties depend on both salinity and calcium, an effective salinity C_{SE} is defined. When the surfactant is non-ionic, the effective salinity is given as

$$C_{SE} = (C_5 - C_6 + \beta C_6)/C_1 \quad (3.14)$$

where $(C_5 - C_6)$ equals the monovalent cation and β is a weighting factor which accounts for the difference in effectiveness between monovalent and divalent cations.

If the surfactant is anionic

$$C_{SE} = (C_3 - C_8 + C_5 - C_6 + \beta C_6)/C_1 \quad (3.15)$$

where C_8 is the calcium-surfactant complex^{8,46} concentration and $(C_3 - C_8 + C_5 - C_6)$ equals the monovalent cation.

When only sodium ion is considered to exist as a cation, C_6 can be used as a tracer by setting β equal to unity, instead of setting C_6 equal to zero.

Although not used in this study, it should be noted that the definition of C_{SE} has been generalized²⁷ to include the surfactant

and alcohol dilution effects.^{1,48-50} Also, in general β for polymer is different from β for surfactant. Just recently, Fil⁶⁵ has implemented Hirasaki's cation exchange-micelle model⁶⁶ which can be used rather than the "complex model" referred to above. Electrolytes are then no longer uniformly distributed.

3.2.2 Phase behavior

In this section, equations required to calculate phase concentrations and saturations are presented. The independent variables here are C_{SE} , C_1 , C_2 , and C_7 . Since adsorbed surfactant is not considered to affect phase behavior, total concentrations add as follows:

$$C_1 + C_2 + C_3 + C_7 = 1 \quad (3.16)$$

When a phase diagram is considered, C_3 and C_7 are summed up to make a single pseudo-component. Thus the pseudo ternary diagram concept is employed.

Although the simulator is designed to deal with Type II(-), Type III, and Type II(+) phase environments, only equations for the Type II(-) and Type II(+) phase environments are presented. When Type III phase environments arise, coordinate rotation is performed and plait points for both two-phase nodes and invariant point are moved continuously according to salinity.³⁷

(1) Binodal curve and distribution curve

For Type II(-) or Type II(+) phase environment, the Hand equation⁵¹ is applied. Regardless of phase number j , the composition corresponding to the point on the binodal curve satisfies the

equation below:

$$\frac{C_{3j}}{C_{1j}} = A \left(\frac{C_{3j}}{C_{2j}} \right)^B \quad (3.17)$$

where parameter A is a function of salinity and is discussed later in more detail. Parameter B is taken to be a constant of minus unity, which yields a symmetric binodal curve, in all the subsequent discussion. The volume fractions must add to one for the pseudo-ternary, so

$$C_{1j} + C_{2j} + C_{3j} = 1 \quad (3.18)$$

Combining equations (3.17) and (3.18) with $B = -1$,

$$C_{3j} = \frac{1}{2} \left(-AC_{2j} + \sqrt{(AC_{2j})^2 + 4AC_{2j}(1 - C_{2j})} \right) \quad (3.19)$$

$$C_{1j} = 1 - C_{2j} - C_{3j} \quad (3.20)$$

In addition to equation (3.17) the concentrations of the two equilibrated phases satisfy the following Hand equations:

$$\left(\frac{C_{32}}{C_{22}} \right) = E \left(\frac{C_{31}}{C_{11}} \right)^F \quad (3.21)$$

Here the definition of phase number is different from that mentioned before only for convenience. Since only two phases are considered, left of the plait point is called phase 1 and the right of

the plait point is phase 2. Calculation of parameter E as well as parameter A is given later. Parameter F is taken to be unity in all subsequent discussion.

(2) Parameter estimation

Parameter A is calculated based on the set of three input parameters, C_{3MAX0} , C_{3MAX1} , and C_{3MAX2} , which are physically the maximum height of the binodal curve at $C_{SEN} = 0, 1$, and 2 , respectively. Here C_{SEN} is normalized salinity: salinity divided by optimal salinity. In this model, the optimal salinity is defined as the salinity that yields a Type III phase environment and an oil concentration at the invariant point of 0.5 . First, parameters A at the three salinities are calculated

$$A_k = \left(\frac{2C_{3MAXk}}{1 - C_{3MAXk}} \right)^2 \quad k = 0, 1, 2 \quad (3.22)$$

Then linear interpolation with respect to C_{SEN} is applied.

$$A = A_0 + (A_1 - A_0)C_{SEN} \quad C_{SEN} \leq 1 \quad (3.23a)$$

$$A = A_1 + (A_2 - A_1)(C_{SEN} - 1) \quad C_{SEN} \geq 1 \quad (3.23b)$$

Parameter E can be obtained from the location of the plait point. Since at the plait point the two phases are exactly the same, equation (3.21) is rewritten:

$$\frac{C_{3p}}{C_{2p}} = E \frac{C_{3p}}{C_{1p}} \quad (3.24)$$

where the subscript p indicates the plait point. Then, solving (3.24) for E gives

$$E = \frac{C_{1p}}{C_{2p}} \quad (3.25)$$

Since the plait point is on the binodal curve, equations (3.19) and (3.20) are applicable. Thus only C_{2p} and A are required to calculate parameter E.

For Type II(+) and Type II(+) phase environments, the oil concentration at the plait point (C_{2p}) is assumed not to change with salinity, while C_{1p} and C_{3p} change according to equations (3.19) and (3.20). The value C_{2p} for both Type II(-) and Type II(+) phase environments must be given as input data.

(3) Calculation of phase concentrations and saturations

Suppose C_{SE} and the total concentrations as well as all input parameters are given and the phase behavior environment is Type II(+) or Type II(-). Then the equations used and unknowns involved are summarized in Table 3.1, which indicates that we need one more equation to solve the system of equations. Since the two-phase compositions are located on the tie line which goes through total composition (see Figure A.1), equation (3.26) below must be satisfied.

$$\frac{C_{32} - C_{31}}{C_{22} - C_{21}} = \frac{C_{32} - C_3}{C_{22} - C_2} \quad (3.26)$$

where the definition of phase number is the same as the one used in equation (3.21). Now the number of equations and the number of unknowns are balanced, which means the equations are solvable somehow. First, A , C_{3p} , C_{1p} , and E are calculated explicitly. Then some iterative method is used to solve equations (3.18), (3.19) and (3.21) for all C_{ij} , if the plait point is not located at the corner. If the plait point is at the corner, every unknown can be solved explicitly, since the composition of the excess phase is known.

Once the composition of both phases is obtained, the saturation of each phase is calculated from overall material balance.

$$C_i = S_1 C_{i1} + S_2 C_{i2} \quad i = 1, 2, 3 \quad (3.27)$$

3.2.3 Adsorption

The adsorption isotherms for both surfactant and polymer are Langmuir-type⁵²

$$\bar{C}_i = \frac{a_i C_{ij}}{1 + b_i C_{ij}^*} \quad i = 3 \text{ or } 4 \quad (3.28)$$

where C_{ij}^* refers to the concentration of component i in the phase richest in component i . Or

$$C_{ij}^* = \max_{j=1,3} (C_{ij}) \quad i = 3 \text{ or } 4 \quad (3.29)$$

Parameters a_i and b_i should be determined from experimental data.

b_i is a constant, while a_i can be a function of salinity, which allows adsorption to be salinity dependent.

$$a_i = a_{i1} + a_{i2} C_{SE} \quad (3.30)$$

In subsequent example calculations, a_4 was assumed to be a constant, which makes polymer adsorption salinity independent. Further assumptions are made as follows. Surfactant adsorption is reversible with salinity but irreversible with surfactant concentration. Polymer adsorption is irreversible.

3.2.4 Phase viscosity

A new generalized viscosity model³⁷ was used.

$$\mu_j = C_{1j} \mu_w e^{\alpha_1 (C_{2j} + C_{3j})} + C_{2j} \mu_o e^{\alpha_2 (C_{1j} + C_{3j})} + C_{3j} \alpha_3 e^{\alpha_4 C_{1j} + \alpha_5 C_{2j}} \quad (3.31)$$

where

μ_w = viscosity of water without polymer

μ_o = viscosity of oil

and the α parameters were assumed to be constants.

When polymer is present in the phase considered, μ_w is replaced by μ_p , which accounts for the effect of polymer. First the

concentration of polymer and salinity is taken into account.³⁹

$$\mu_p = \mu_w R_k [1 + (A_{p1} C_{4j} + A_{p2} C_{4j}^2 + A_{p3} C_{4j}^3) C_{SE}^{s_p}] \quad (3.32)$$

where

$$R_k = 1 + \frac{(R_{kmax} - 1) b_p C_{4j}}{1 + b_p C_{4j}} \quad (3.33)$$

A_{p1}, A_{p2}, A_{p3} = constant coefficient

s_p = constant exponent

R_k = permeability reduction factor

R_{kmax} = maximum value of R_k

b_p = constant coefficient

In equation (3.32), the permeability reduction factor is multiplied to increase viscosity rather than decreasing permeability. From the viewpoint of mobility, they have the same effect. The permeability reduction is modeled as permanent (irreversible).

3.2.5 Interfacial tension

A set of two empirical equations presented by Reed and Healy¹ are used to calculate interfacial tensions (IFT's)

$$\log \gamma_{wm} = G_{12} + \frac{G_{11}}{G_{13} (C_{13}/C_{33}) + 1} \quad (3.34a)$$

$$\log \gamma_{mo} = G_{22} + \frac{G_{21}}{G_{23} (C_{23}/C_{33}) + 1} \quad (3.34b)$$

where

γ_{wm} = interfacial tension between aqueous and microemulsion
phase

γ_{mo} = interfacial tension between microemulsion and oleic
phase

and parameters (G's) must be obtained from experimental data.

When the phase environment is Type II(-), only equation (3.34b) is used while type II(+) requires only equation (3.34a). If phase environment is Type III and three phases coexist, both equations (3.34a) and (3.34b) are used to obtain two interfacial tensions. As concerns either Type II(-) node or Type II(+) node of Type III, one of equations (3.34) is used in a similar way to Type II(-) or Type II(+) phase environment. Several examples of the effect of the G's on the calculated interfacial tension are shown in Figures 3.3 through 3.6. In these figures, the solubilization parameter designates the ratio C_{13}/C_{33} or C_{23}/C_{33} in equations (3.34).

3.2.6 Trapping function and residual phase saturation

Several authors⁵³⁻⁵⁵ have shown the dependence of residual phase saturation on the capillary number, which represents the ratio of viscous force to capillary force. Figure 3.7 shows the typical example presented by Gupta and Trushenski,⁵⁵ which suggests the applicability of equations below for the regions where residual saturation changes

$$S_{jr} = a_j + b_j \log \left(\frac{\Delta p k}{L \gamma} \right) \quad (3.35)$$

where S_{jr} is the residual saturation of phase j , which can be either wetting phase or non-wetting phase. And

$a, b = \text{constant}$

$\Delta P/L = \text{pressure gradient}$

$k = \text{absolute permeability}$

$\gamma = \text{interfacial tension}$

From Darcy's law, the capillary number in equation (3.35) can be expressed in an alternative way for multiphase flow

$$\frac{\Delta p k}{L \gamma} = \frac{q}{A \lambda_{rT} \gamma} \quad (3.36)$$

where

$q = \text{volumetric flow rate}$

$A = \text{cross-sectional area}$

$$\lambda_{rT} = \sum_j (k_{rj} / \mu_j)$$

For a given flow rate with constant area, substitution of equation (3.36) into equation (3.35) yields

$$S_{jr} = a'_j - b_j \log(\lambda_{rT} \gamma) \quad (3.37)$$

Since, in a water-oil system with no chemical, the residual saturation of water (S_{1rw}) and oil (S_{2rw}) can be considered to be constant,

equations (3.35) can be rewritten

$$S_{jr} = S_{jrw} \{1 + T_{j1}[\log(\lambda_{rT}\gamma) + T_{j2}]\} \quad (3.38)$$

where S_{jrw} equals to S_{1rw} for wetting phase and equals to S_{2rw} for non-wetting phase. Parameters T 's in equation (3.38) depend on fluid/rock properties such as wettability and have to be determined from experimental data. When only two phases exist, there is only one interfacial tension considered, and it is substituted into equations (3.38) for both wetting phase and non-wetting phase to calculate residual saturations. Meanwhile for the case where three phases coexist, phase trapping behavior is still poorly understood. Although much more experimental work and prudent investigation is being expected in this area, a few models have been suggested and will be discussed later.

Since equations (3.38) are applied only to the region where residual saturations change as shown in Figure 3.7, the residual is set to the water-oil (no-surfactant) value when the calculated residual exceeds the water-oil value. If the calculated residual is negative, the residual is set to zero. Furthermore, as a special feature of chemical flooding, saturations can become less than the residual due to the phase behavior (partitioning or mass transfer). In such cases the residual saturations are set to the saturations calculated after the "flash" calculation.

3.2.7 Relative Permeability

The relative permeability model used in this research was modified from the one used in the original model. The basic idea is to make relative permeabilities (k_{rj} 's) approach the proper limits when surfactant is involved. In this section, only the equations for two-phase relative permeabilities are presented. For three-phase flow, three different models will be introduced and discussed in the next chapter. When there exist only two phases, the requirements are

(1) k_{rj} 's approach their water-oil (no surfactant) values as the capillary number decreases

(2) k_{rj} 's approach their respective phase saturations as capillary number increases

There are several cases which involve only two phases: surfactant free, Type II(-), or Type II(+), phase environment, and either of the two phase nodes of the Type III phase environment. In these cases, one phase can be identified as wetting and the other as non-wetting, presuming that one phase preferentially wets the rock surface.

The assumed relative permeabilities are

$$k_{rj} = k_{rj}^0 \left(\frac{S_j - S_{jr}}{1 - S_{jr} - S_{j'r}} \right)^{e_j} \quad (3.39)$$

$j \neq j'$

where

S_j = saturation of phase j

S_{jr} = residual saturation of phase j

k_{rj}^0 = end point relative permeabilities

(k_{rj} -value at other phase's residual saturation to phase j)

e_j = "curvature" of relative permeability curve of phase j in reduced saturation space

Again in equations (3.39), phase j can be either wetting or non-wetting phase. When phase j is wetting phase, phase j' is non-wetting phase (and vice-versa). S_j 's are obtained from equation (3.27) in phase behavior calculation while S_{jr} 's are calculated using equations (3.38).

When a reservoir is preferentially water wet, the aqueous phase is assumed to be the wetting phase compared with the microemulsion phase in the Type II(+) phase environment, whereas the microemulsion phase wets and the oleic phase is non-wetting in the Type II(-) environment.

In order to make relative permeabilities approach the proper limits, linear interpolation of end points and curvatures of equations (3.39) are performed based on the change in the residual phase saturations as follows:

$$k_{rj}^0 = k_{rjw}^0 + \frac{S_{j'rw} - S_{j'r}}{S_{j'r}} (k_{rjc}^0 - k_{rjw}^0) \quad j \neq j' \quad (3.40)$$

$$e_j = e_{jw} + \frac{S_{j'rw} - S_{j'r}}{S_{j'r}} (e_{jc} - e_{jw}) \quad j \neq j' \quad (3.41)$$

where subscript w designates a water-oil (no surfactant) quantity and subscript c the infinite capillary number value. Although all values with subscript c are usually considered to be unity, they are left to be specified in input data for flexibility.

3.2.8 Other features

In addition to the features which have been described so far, the simulator involves several other features. Since such features are not used in this research and they are discussed elsewhere in detail^{38,39,47}, only the list of such features is given here.

- (1) Inaccessible Pore Volume
- (2) Shear rate effect on polymer
- (3) Ion exchange
- (4) Surfactant complex
- (5) Alcohol effect
- (6) Dilution effect

3.3 Solution Procedure

Summarizing the functional relations described so far and governing continuity equations, the interdependence of the major variables is shown in Figure 3.8. All variables are considered at the same time level except the calculation of \tilde{C}_1 from its

time derivative, which is indicated by dashed line.

Figure 3.9 illustrates the solution procedure employed in the simulator. Adsorption of both polymer and chemical are obtained explicitly using the phase concentration at the old time level. Residual saturations S_{jr} are first calculated based on total relative mobility of old time level, then iteration is performed, if necessary, with the secant method.

To start the computation, boundary conditions as well as all parameters necessary have to be given. The step-by-step computational procedure is outlined as follows. Since the features mentioned in Section 3.3.8 were not used in this research, they are excluded from the procedure. For each grid block,

- (1) Based on initial condition, calculate f_j , C_{ij} and λ_{rT}
- (2) Calculate \tilde{C}_i at the new time level by solving continuity equations (3.1) or (3.5)
- (3) Calculate C_{SE} from equation (3.14)
- (4) Calculate chemical adsorption \bar{C}_3 , if needed, from equation (3.28). C_{3j}^* of old time level is used.
- (5) Calculate C_i of equation (3.16) by excluding \bar{C}_3 .
- (6) Calculate \bar{C}_4 , if necessary, from equation (3.28). C_{4j}^* of old time level is used
- (7) Calculate C_{ij} based on C_i and C_{SE} with binodal curve and distribution curve equations.

(8) Calculate S_j from C_i and C_{ij} making use of equation (3.27)

(9) Calculate γ from equation (3.34)

(10) Calculate μ_j from equations (3.31) through (3.33)

(11) Calculate S_{jr} from equation (3.38). λ_{rT} of old time level is used at the first time. Then λ_{rT} obtained at step (13) is used when iterated.

(12) Calculate k_{rj} from equations (3.39) through (3.41)

(13) Calculate λ_{rj} , λ_{rT} and f_j based on k_{rj} and μ_j

(14) Compare new λ_{rT} with old λ_{rT} . If the relative difference is larger than some specified value, go back to step (11) and repeat calculation. Secant method is used to obtain next estimate. If the difference is small enough, go to step (2) and start new time level calculation.

Table 3.1. Equations used to calculate phase concentrations.
 Note: C_{3MAX0} , C_{3MAX1} , C_{3MAX2} and C_{2p} are input parameters.

Equation No.	Equation	Number of Equations	Unknowns	Number of Unknowns
(3.18)	$C_{ij} + C_{2j} + C_{3j} = 1 \quad j = 1, 2 \text{ or } p$	3	C_{ij}, C_{1p}, C_{3p}	8
(3.19)	$C_{3j} = \frac{1}{2} \left(-AC_{2j} + \sqrt{(AC_{2j})^2 + 4AC_{2j}(1 - C_{2j})} \right)$ $j = 1, 2, \text{ or } p$	3	A	1
(3.21)	$\frac{C_{32}}{C_{22}} = E \frac{C_{31}}{C_{11}}$	1	E	1
(3.22)	$A_k = \left(\frac{2C_{3MAXk}}{1 - C_{3MAXk}} \right)^2 \quad k = 0, 1, 2$	3	A_k	3
(3.23)	$A = A_0 + (A_1 - A_0)C_{SEN} \quad C_{SEN} \leq 1$ $A = A_1 + (A_2 - A_1)(C_{SEN} - 1) \quad C_{SEN} \geq 1$	1	-	0
(3.25)	$E = C_{1p}/C_{2p}$	1	-	0
		12		13

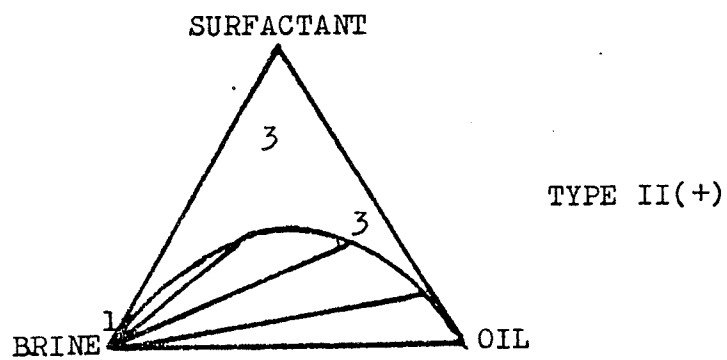
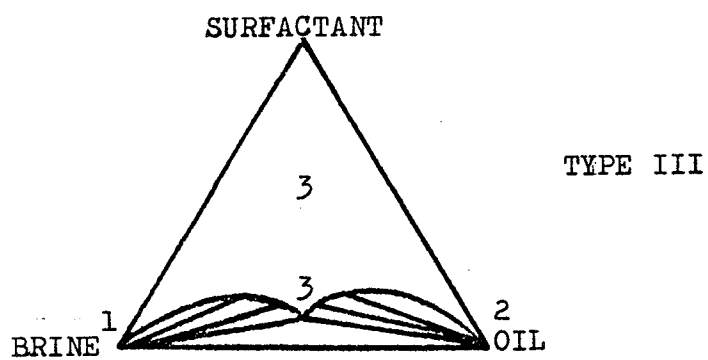
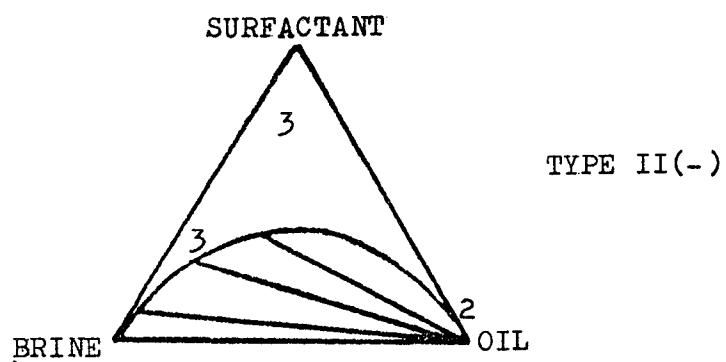


Figure 3.2. Definition of phase number.

SYMBOL	G11	G12	G13
PLUS	5.0000	-7.0000	.1000
BOX	6.0000	-7.0000	.1000
OCTAGON	7.0000	-7.0000	.1000
TRIANGLE	8.0000	-7.0000	.1000

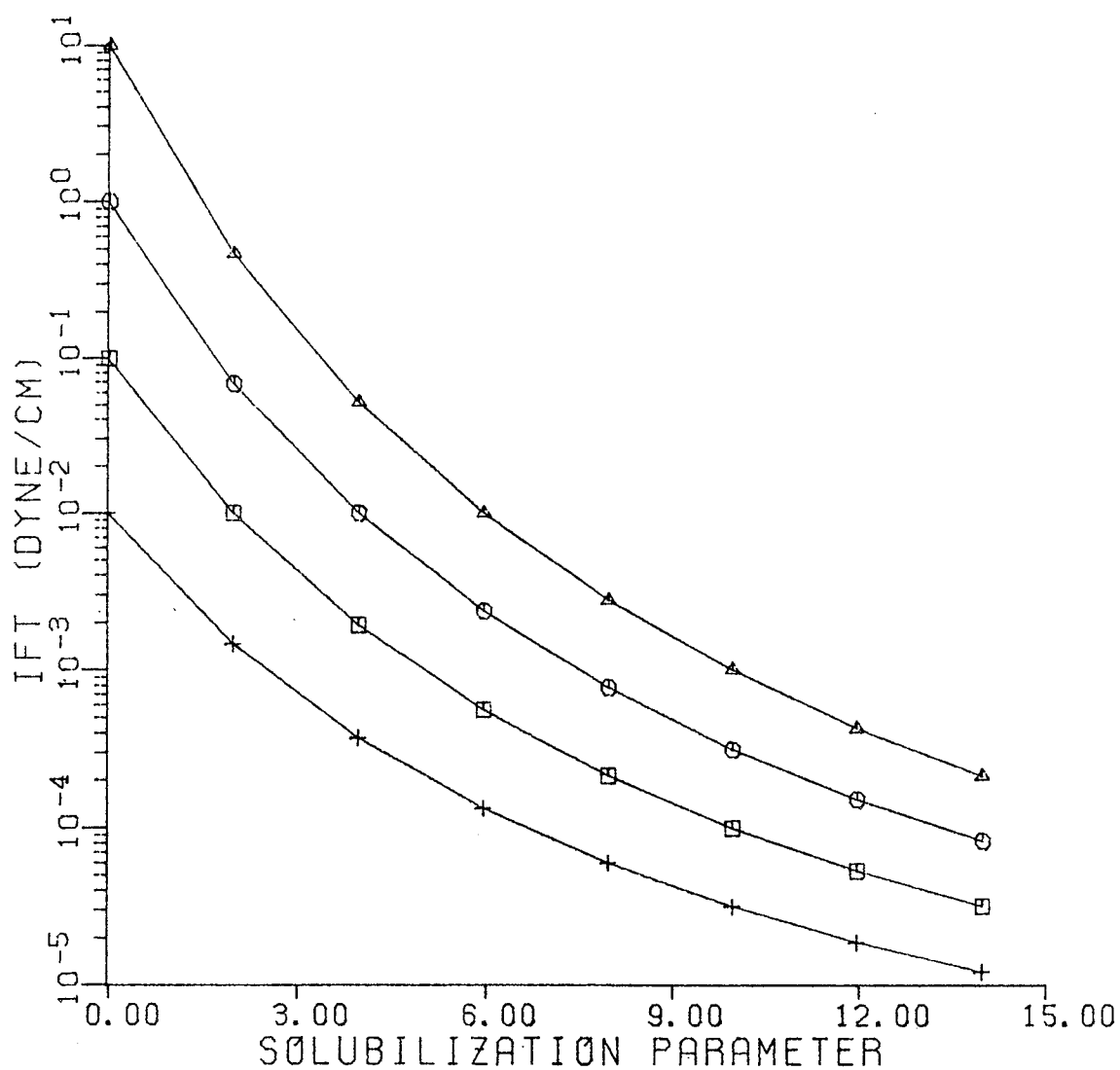


Figure 3.3. Effect of G parameters on calculated IFT curve.

SYMBOL	G11	G12	G13
PLUS	6.0000	-5.0000	.1000
BOX	6.0000	-6.0000	.1000
OCTAGON	6.0000	-7.0000	.1000
TRIANGLE	6.0000	-8.0000	.1000

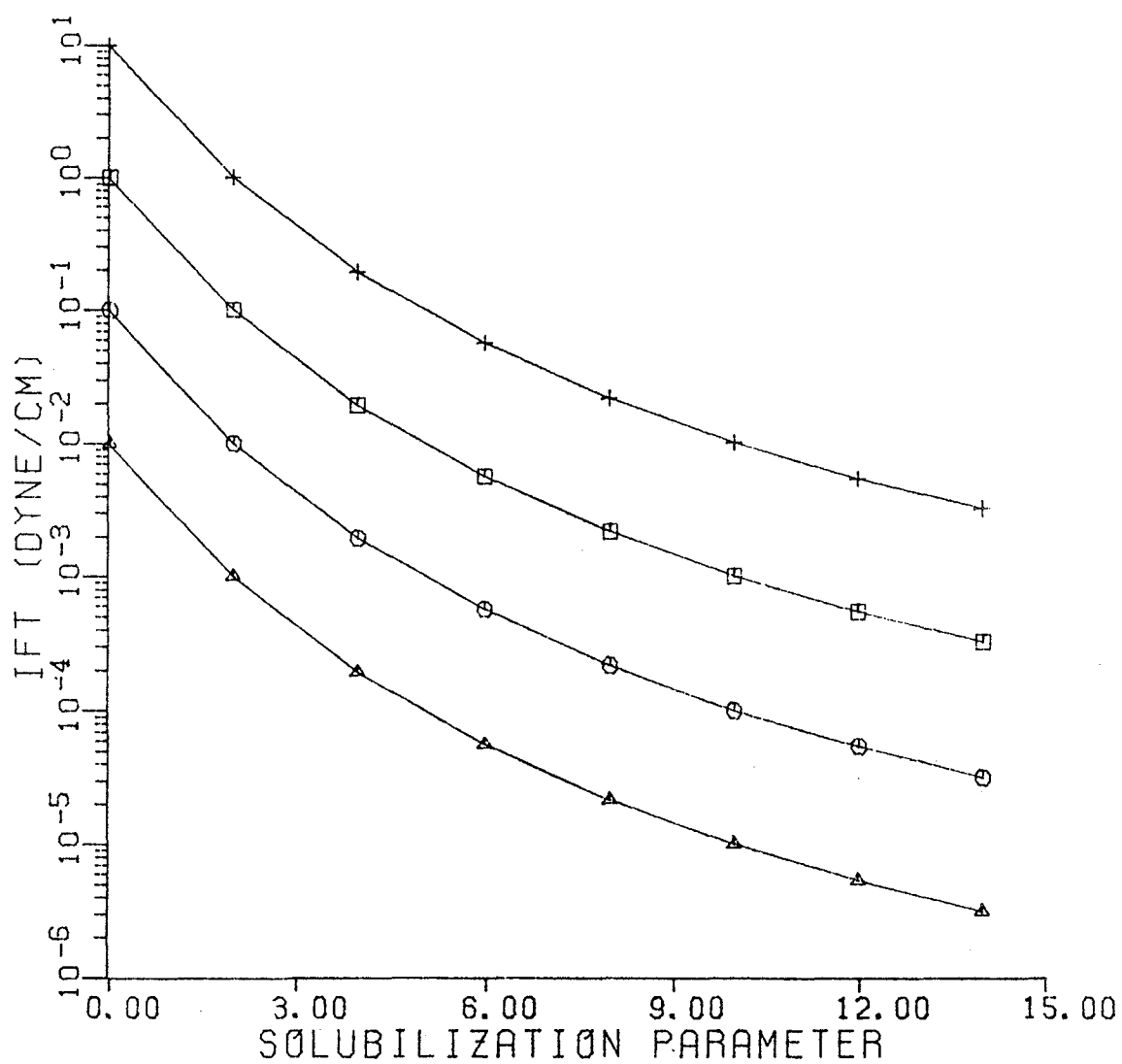


Figure 3.4. Effect of G parameters on calculated IFT curve.

SYMBOL	G11	G12	G13
PLUS	6.0000	-7.0000	.0100
BOX	6.0000	-7.0000	.0500
OCTAGON	6.0000	-7.0000	.1000
TRIANGLE	6.0000	-7.0000	.2000

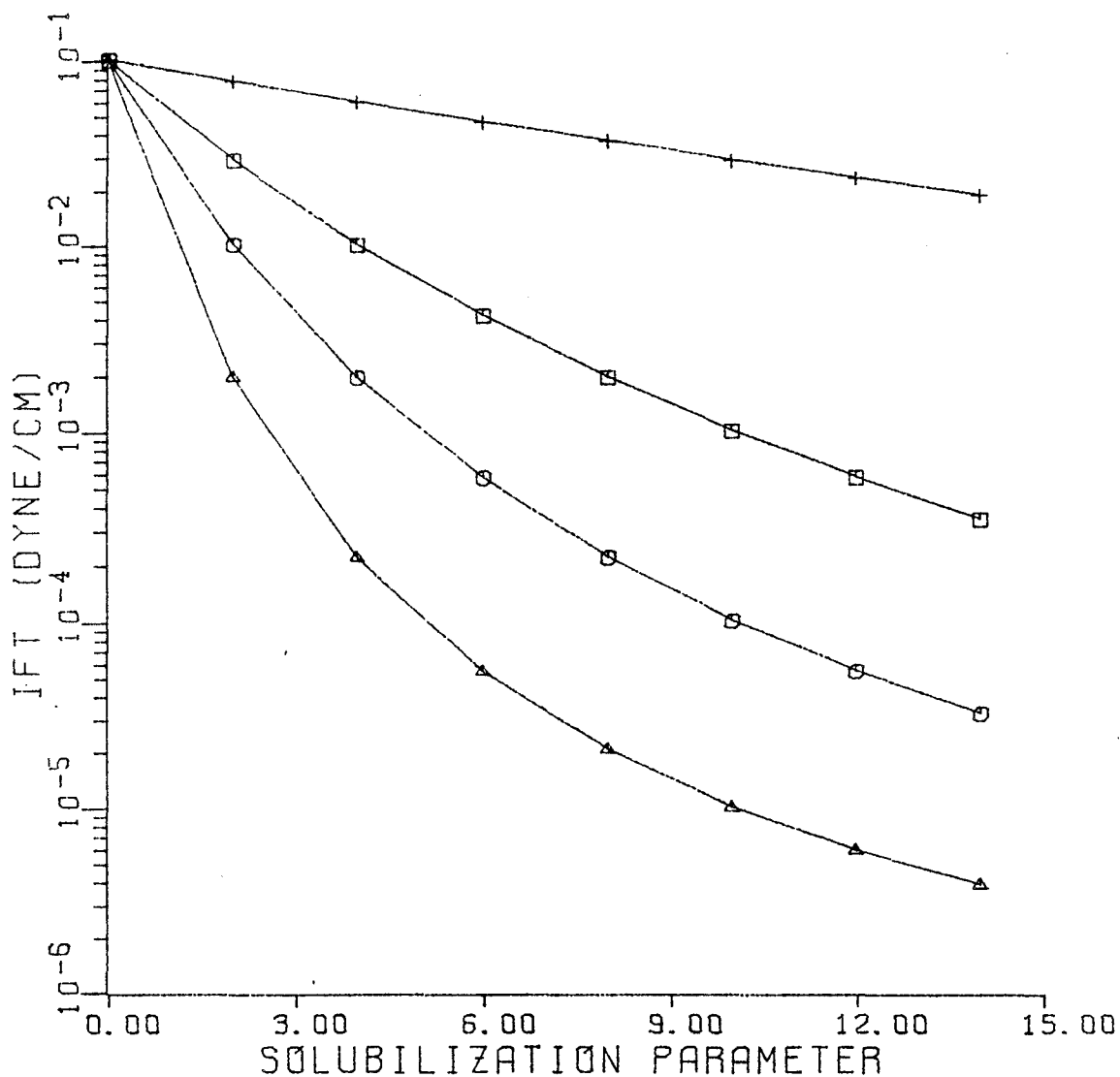


Figure 3.5. Effect of G parameters on calculated IFT curve.

SYMBOL	G11	G12	G13
PLUS	6.0000	-7.0000	.1000
BOX	8.0000	-9.0000	.1000
OCTAGON	10.0000	-11.0000	.1000
TRIANGLE	12.0000	-13.0000	.1000

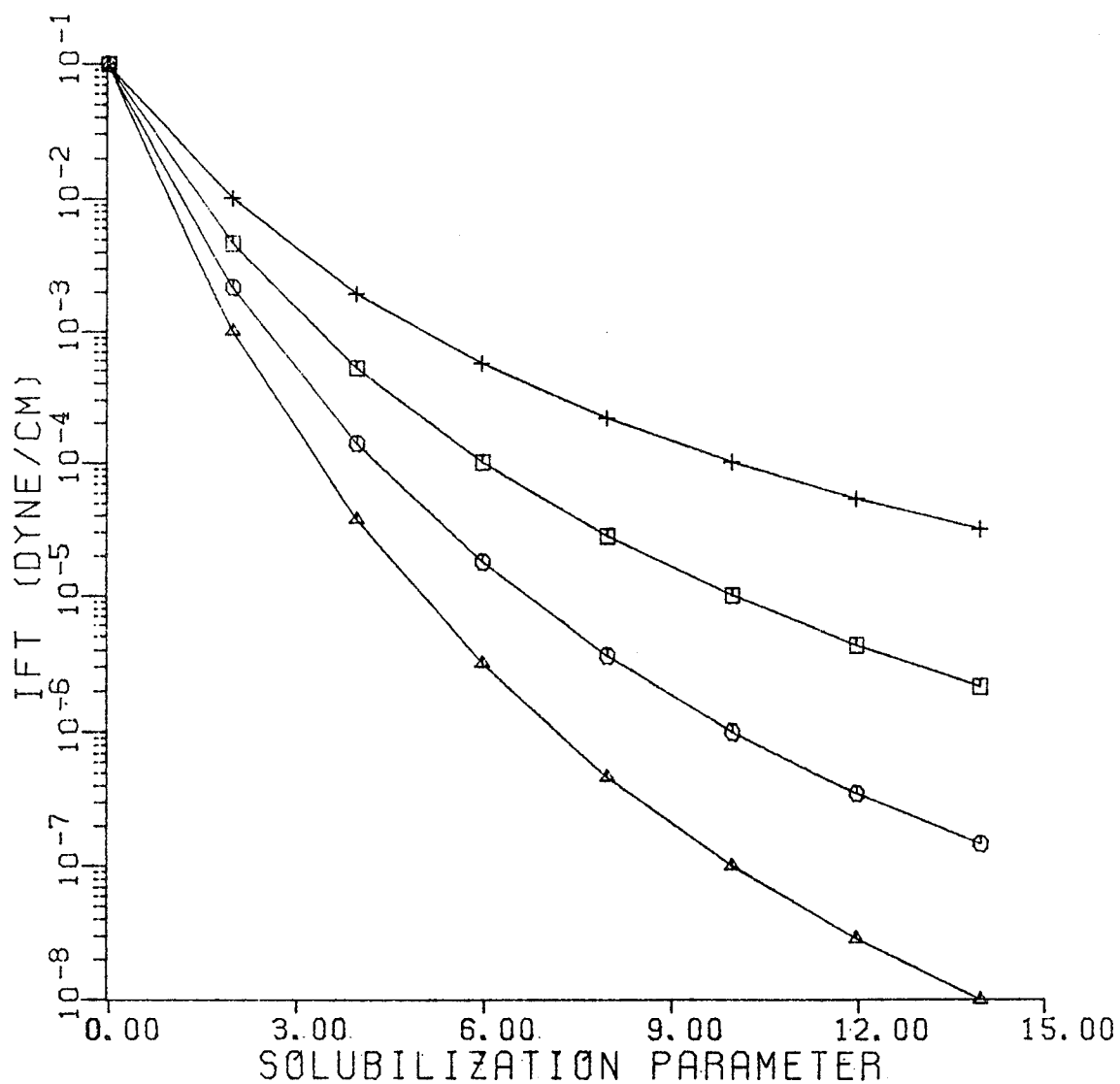


Figure 3.6. Effect of G parameters on calculated IFT curve.

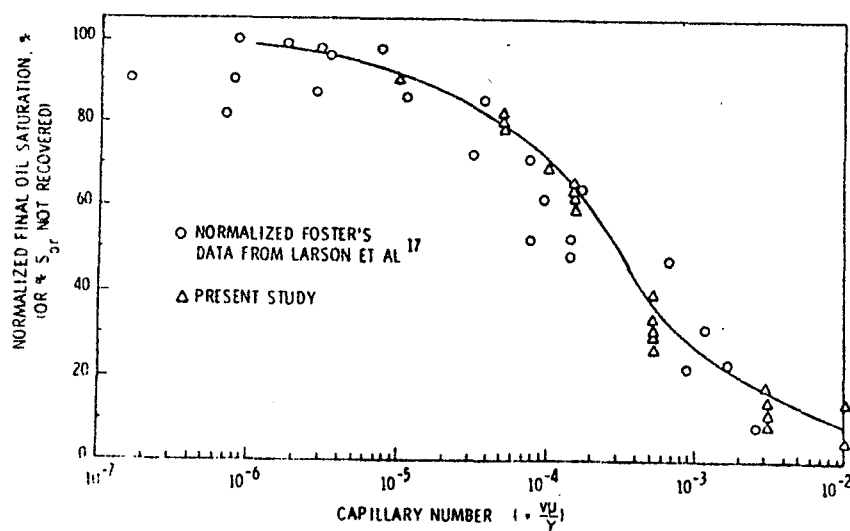
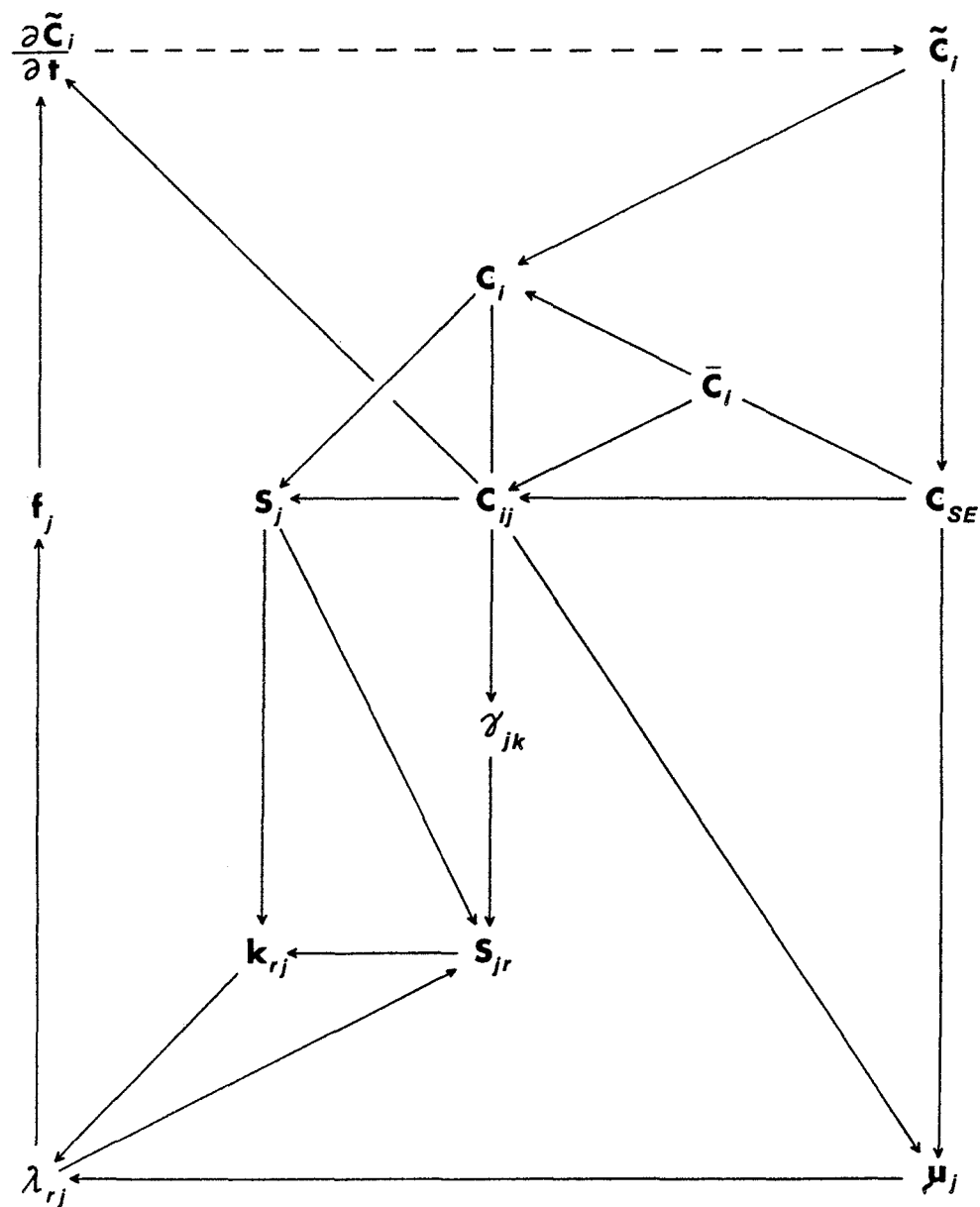


Figure 3.7. Normalized residual saturation versus capillary number.⁵⁵



$A \rightarrow B$ means B depends on A

Figure 3.8. Interdependence of variables (implicit scheme) without shear rate effect on polymer solution.

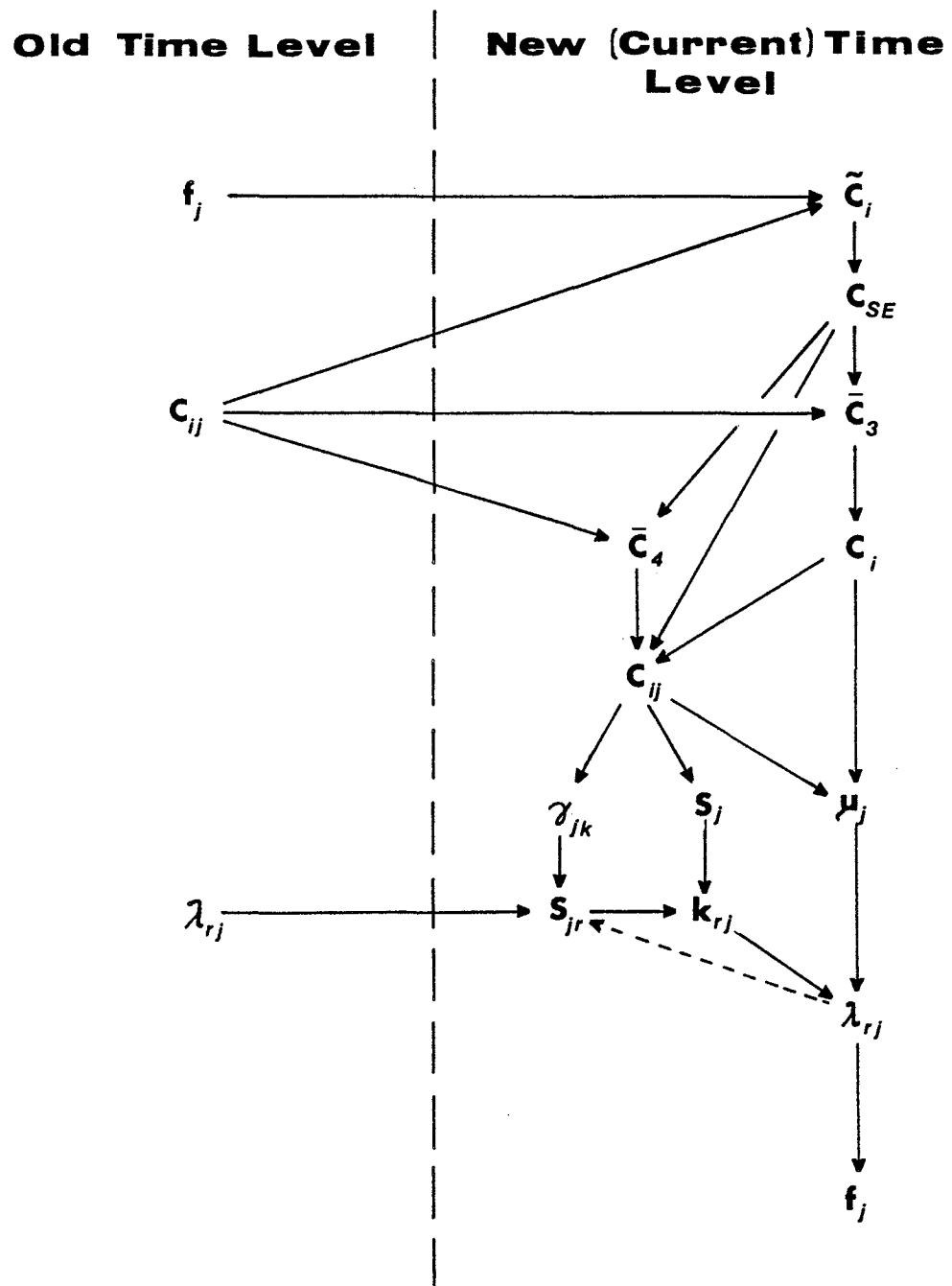


Figure 3.9. Solution procedures employed in the simulator.

CHAPTER IV

THREE PHASE FLOW MODEL

When salinity is in a certain range, phase behavior environment is called Type III (see Appendix A) and its phase diagram can involve a three phase region. When three phases coexist, little is known about the trapping of each phase and their flow character. However, the modeling of such three phase flow is necessary since the process is usually used where the lowest interfacial tensions are achieved, which is in the three phase region. Thus, several authors have developed models based on various assumptions.

In this chapter, three examples of such three phase flow models are introduced and comparisons are presented.

4.1 Pope's model

The first example is the one used by Pope in his simulator.⁵ He assumed another trapping function similar to equation (3.38) for the microemulsion phase. To calculate its residual saturation, the smaller value of γ_{wm} and γ_{mo} calculated from equations (3.34) is taken. Residual saturations for excess oil and excess water are similar to the two phase model. γ_{wm} is used to calculate residual water and γ_{mo} for residual oil. Then k_r of each phase is given by

$$k_{r1} = k_{r1}^0 \left(\frac{S_1 - S_{1r}}{1 - S_{1r} - S_{2r} - S_{3r}} \right) \quad (4.1a)$$

$$k_{r2} = k_{r2}^0 \left(\frac{s_2 - s_{2r}}{1 - s_{1r} - s_{2r} - s_{3r}} \right) \quad (4.1b)$$

$$k_{r3} = k_{r3}^0 \left(\frac{s_3 - s_{3r}}{1 - s_{1r} - s_{2r} - s_{3r}} \right)^{e_3} \quad (4.1c)$$

where subscripts 1, 2, and 3 designate water, oil, and microemulsion phase respectively. k_{r1}^0 and k_{r2}^0 are given by equations (3.40).

4.2 Hirasaki's model

Another model was presented by Hirasaki⁹. He calculated the residual saturation of each phase based on a physical idea, which is shown in Figure 4.1. In describing the model, an assumption is made here that a preferentially water wet reservoir is considered. This assumption is just to make explanation easier and the generality of the model is not affected by the assumption.

Figure 4.1a leads to equation (4.2) which describes the trapping of excess oil and microemulsion phase by the excess water phase

$$s_{3r} + s_{2r} = f(\gamma_{wm}) \quad (4.2)$$

where $f(\gamma)$ is the non-wetting phase trapping function, which is identical to the right hand side of equation (3.38) in the simulator.

Figure 4.1b shows the trapping of excess water and microemulsion

phases by the excess oil phase

$$S_{3r} + S_{1r} = g(\gamma_{mo}) \quad (4.3)$$

where $g(\gamma)$ is the wetting phase trapping function which is given as equation (3.38)

From Figures 4.1c and 4.1d

$$S_{2r} = f(\gamma_{mo}) \quad (4.4)$$

$$S_{1r} = g(\gamma_{wm}) \quad (4.5)$$

After evaluating equations (4.2) through (4.5), the residual saturation of each phase is determined as follows:

$$S_{1r} = \max \begin{cases} g(\gamma_{wm}) \\ g(\gamma_{mo}) - S_3 \end{cases} \quad (4.6a)$$

$$S_{2r} = \max \begin{cases} f(\gamma_{mo}) \\ f(\gamma_{wm}) - S_3 \end{cases} \quad (4.6b)$$

$$S_{3r} = \max \begin{cases} f(\gamma_{wm}) - S_2 \\ 0 \\ g(\gamma_{mo}) - S_1 \end{cases} \quad (4.6c)$$

Then assumed relative permeabilities are

$$k_{rj} = k_{rj}^0 S_{nj}^{e_j} \quad (4.7)$$

$$(j = 1, 2, 3)$$

where

$$S_{n1} = \frac{S_1 - S_{1r}}{1 - f(\gamma_{wm}) - S_{1r}} \quad (4.8a)$$

$$S_{n2} = \frac{S_2 - S_{2r}}{1 - g(\gamma_{mo}) - S_{2r}} \quad (4.8b)$$

$$S_{n3} = \frac{S_3 - S_{3r}}{1 - S_{1r} - S_{2r} - S_{3r}} \quad (4.8c)$$

$$k_{r1}^0 = k_{r1c}^0 - (k_{r1c}^0 - k_{r1w}^0)f(\gamma_{wm})/S_{2rw} \quad (4.9a)$$

$$k_{r2}^0 = k_{r2c}^0 - (k_{r2c}^0 - k_{r2w}^0)g(\gamma_{mo})/S_{1rw} \quad (4.9b)$$

$$k_{r3}^0 = \omega k_{r1}^0 + (1 - \omega)k_{r2}^0 \quad (4.9c)$$

$$e_1 = 1.0 + (e_{1w} - 1.0)f(\gamma_{wm})/S_{2rw} \quad (4.10a)$$

$$e_2 = 1.0 + (e_{2w} - 1.0)g(\gamma_{mo})/S_{1rw} \quad (4.10b)$$

$$e_3 = \omega e_1 + (1 - \omega)e_2 \quad (4.10c)$$

$$\omega = S_{2T}/(S_{1T} + S_{2T}) \quad (4.11)$$

$$S_{1T} = \min(S_{1r}, S_1) \quad (4.12a)$$

$$S_{2T} = \min(S_{2r}, S_2) \quad (4.12b)$$

4.3 Lake's model³⁷

The basic philosophy employed in the model is that the intermediate wetting phase becomes the wetting phase when the original wetting phase is absent (and vice versa). Hence he first introduced a simple interpolating function

$$G(S_1, S_2) = \frac{S_2(1 - S_1)}{S_1 + S_2} \quad (4.13)$$

This function gives $G = 0$ when $S_2 = 0$ and $G = 1$ when $S_1 = 0$.

Then the microemulsion residual saturation S_{3r} is determined

$$S_{3r} = S_{2r} + G(S_{1r} - S_{2r}) \quad (4.14)$$

where S_{1r} and S_{2r} are given by equations (3.38).

Assumed relative permeabilities are in the same form as Hirasaki's model.

$$k_{rj} = k_{rj}^0 S_{nj}^{ej} \quad (j = 1, 2, 3) \quad (4.7)$$

where

$$S_{nj} = \frac{S_j - S_{jr}}{1 - S_{1r} - S_{2r} - S_{3r}} \quad j = 1, 2, 3 \quad (4.15)$$

$$k_{r1}^0 = k_{r1c}^0 - (k_{r1c}^0 - k_{r1w}^0) s_{2r}/s_{2rw} \quad (4.16a)$$

$$k_{r2}^0 = k_{r2c}^0 - (k_{r2c}^0 - k_{r2w}^0) s_{1r}/s_{1rw} \quad (4.16b)$$

$$k_{3r}^0 = Gk_{r1}^0 + (1 - G)k_{r2}^0 \quad (4.16c)$$

$$e_1 = 1.0 + (e_{1w} - 1.0) s_{2r}/s_{2rw} \quad (4.17a)$$

$$e_2 = 1.0 + (e_{2w} - 1.0) s_{1r}/s_{1rw} \quad (4.17b)$$

$$e_3 = Ge_1 + (1 - G)e_2 \quad (4.17c)$$

Considering the fact that saturation can become less than residual saturation due to phase behavior, the residual saturation of each phase is defined

$$S_{jr} = \min(S_j, S_{jr}) \quad j = 1, 2, 3 \quad (4.18)$$

Equation (4.18) is substituted in equations (4.15) through (4.17)

as S_{jr} .

4.4 Comparison of Each Model

Tables 4.1 and 4.2 show the comparison of oil recovery and the amount of surfactant trapped with the different three-phase flow models which have been introduced. Table 4.1 shows the results obtained with 0.1 PV of 3% surfactant slug injection whereas 0.1 PV of 6% surfactant slug was injected for results given in Table 4.2. In each table only the flow model was changed for three different salinities. The same results are plotted as oil recovery versus salinity and trapped surfactant versus salinity in Figures 4.3 and 4.4. Each salinity represents near lower limit (CSEL), middle, and near upper limit (CSEU) of Type III phase behavior environment. The change in salinity affects the shape of multiphase region and interfacial tensions as shown in Figure 4.2. Salinity was kept constant or nearly constant for each run. After surfactant slug injection, 1.9 PV of polymer solution was injected in all runs.

The same input data as is given in Table 6.1a was used except that $G_{13} = G_{23} = 0.05$. No adsorption was considered. All other data are shown in Tables 4.4. No microemulsion phase trapping was considered for Pope's model.

When salinity is near lower limit of Type III (Figure 4.2a), oil recovery is rather low with all models. All surfactant injected was trapped with Lake's model whereas the other two models trap no surfactant.

When salinity is around optimal (Figure 4.2b), all models but Pope's trap surfactant somewhat. Surfactant trapping is rather

low and oil recovery is high with all models.

When salinity is near upper limit of Type III (Figure 4.2c), surfactant trapping is rather high with all models. Pope's and Lake's model give high oil recovery while Hirasuki's model gives lower oil recovery.

Although the difference in oil recovery is rather large, especially when the injected amount of surfactant is small, among the models, Pope's model and Hirasaki's model show similar trend in surfactant trapping to each other. Figure 4.5 shows the histories of total concentration in production for each model with salinity of 0.82 (\approx CSEL) and 3% surfactant slug. Not only oil production but surfactant breakthrough time differs among the models.

Table 4.3 shows the comparison for the cases with salinity gradient. Data set 3S-4 and 3S-5 were used for these runs. 0.1 PV of 3% surfactant slug was injected. Surfactant trapping was almost zero in all runs. The difference in oil recovery among the models is rather small compared with constant salinity runs. Figure 4.6 shows production history of each model with salinity gradient (1.4 - 1.0 - 0.6). Although this figure shows there is a significant difference in surfactant production history, the fact that all models yield high oil recovery can be another reason for designing a micellar flood with a salinity gradient.

Table 4.1 Comparison of oil recovery and surfactant trapping.
0.1 PV of 3% surfactant slug is injected.

		Oil Recovery (%)			Trapped Surfactant (PV)		
Normalized C_{se}		0.82	1.0	1.18	0.82	1.0	1.18
Model	Pope	42.5	78.0	95.0	3.3×10^{-17} (0.0%)	6.0×10^{-12} (0.0%)	2.9×10^{-3} (97%)
	Hirasaki	56.1	91.3	62.0	2.7×10^{-15} (0.0%)	1.4×10^{-4} (4.7%)	2.3×10^{-3} (77%)
	Lake	25.5	96.5	93.7	3.0×10^{-3} (100%)	8.9×10^{-4} (30%)	2.7×10^{-3} (90%)

Table 4.2 Comparison of oil recovery and surfactant trapping.
0.1 PV of 6% surfactant slug is injected.

		Oil Recovery (%)			Trapped Surfactant (PV)		
Normalized C_{SE}		0.82	1.0	1.18	0.82	1.0	1.18
Model	Pope	45.2	88.5	87.0	1.1×10^{-16} (0.0%)	2.6×10^{-10} (0.0%)	5.4×10^{-3} (90%)
	Hirasaki	58.5	94.0	76.0	3.2×10^{-15} (0.0%)	2.8×10^{-4} (4.7%)	4.6×10^{-3} (77%)
	Lake	47.5	97.4	89.4	6.0×10^{-3} (100%)	6.8×10^{-4} (11%)	4.7×10^{-3} (78%)

Table 4.3 Comparison of oil recovery and surfactant trapping
for two different salinity gradients.
0.1 PV of 3% surfactant slug is injected.

Salinity Gradient	1.8-1.0-0.2		1.4-1.0-0.6	
Model	Oil Recovery (%)	Surfactant Trapped (PV)	Oil Recovery (%)	Surfactant Trapped (PV)
Pope	90.9	1.5×10^{-10}	93.0	2.5×10^{-10}
Hirasaki	87.9	1.1×10^{-10}	90.4	1.6×10^{-10}
Lake	84.7	1.1×10^{-10}	86.8	1.6×10^{-10}

Table 4.4a Input data used to compare three phase flow models

Composition of injected slug

Data Set	Slug No.	Slug size	Water (vol.frac.)	Oil (vol.frac.)	Surfactant* (vol.frac.)	Polymer (wt%)	Anion	Tracer
3S	1	0.1	0.97	0.0	0.03	0.10	C5(1)**	1.0
	2	1.9	1.0	0.0	0.0	0.10	C5(2)**	1.0
6S	1	0.1	0.94	0.0	0.06	0.10	C5(1)**	1.0
	2	1.9	1.0	0.0	0.0	0.10	C5(2)**	1.0

*Surfactant is combined with alcohol as an approximation.

**See Table 4.4b.

Table 4.4b. Input data used to compare three phase
flow models

Salinity sequence

Data Set	C51I	C5(1)*	C5(2)
3S-1	0.82	0.7954	0.82
3S-2	1.0	1.0	1.0
3S-3	1.18	1.1446	1.18
6S-1	0.82	0.7708	0.82
6S-2	1.0	1.0	1.0
6S-3	1.18	1.1092	1.18
3S-4	1.8	0.97	0.2
3S-5	1.4	0.97	0.6

C51I = Initial anion concentration in a water phase (normalized)

C5(1) = Anion concentration in surfactant slug (normalized)

C5(2) = Anion concentration in polymer buffer (normalized)

*This is the total concentration. This value must be divided
by water concentration in the slug to obtain effective salinity.

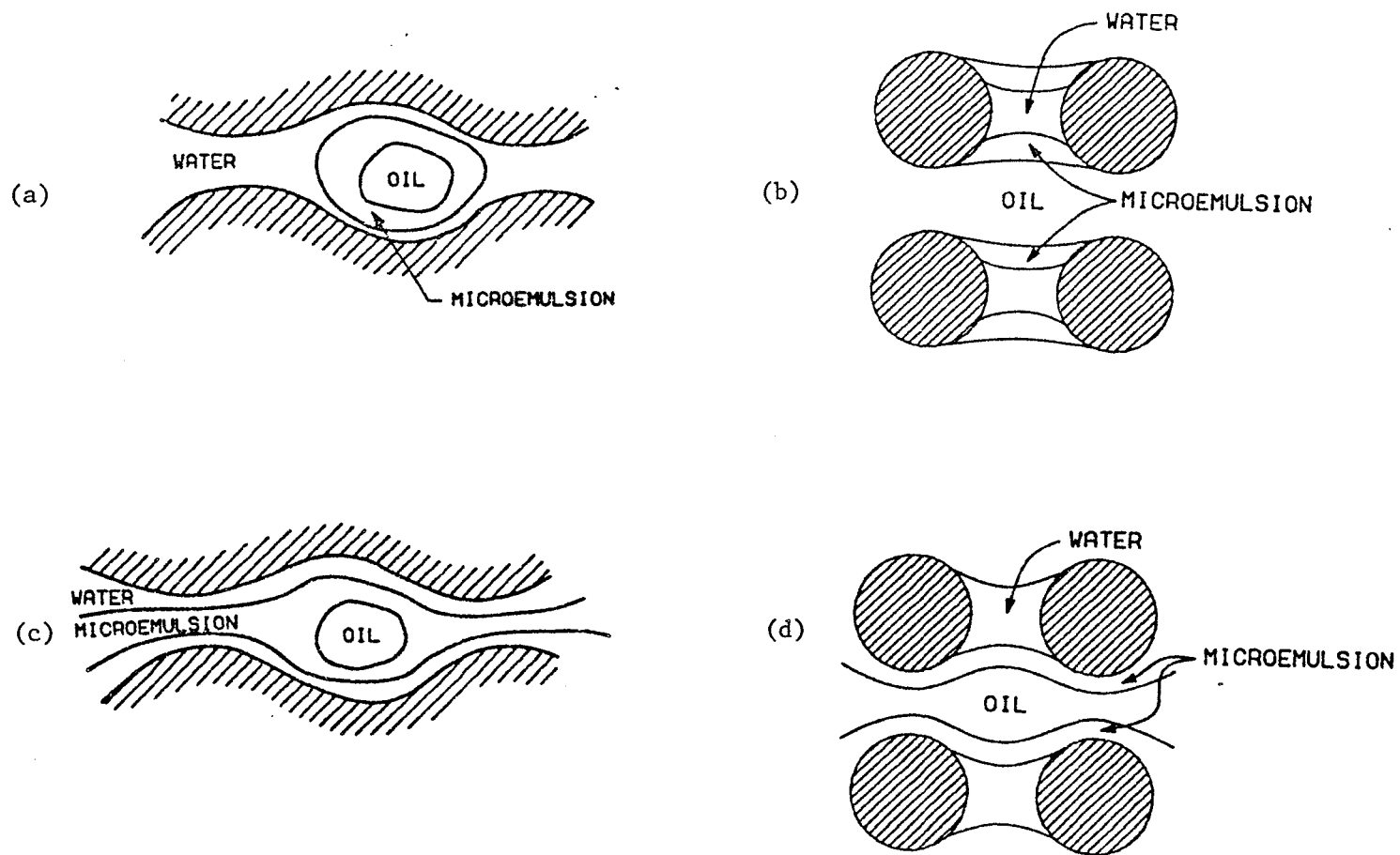


Figure 4.1. Basic idea of phase trapping employed in Hirasaki's model.⁹

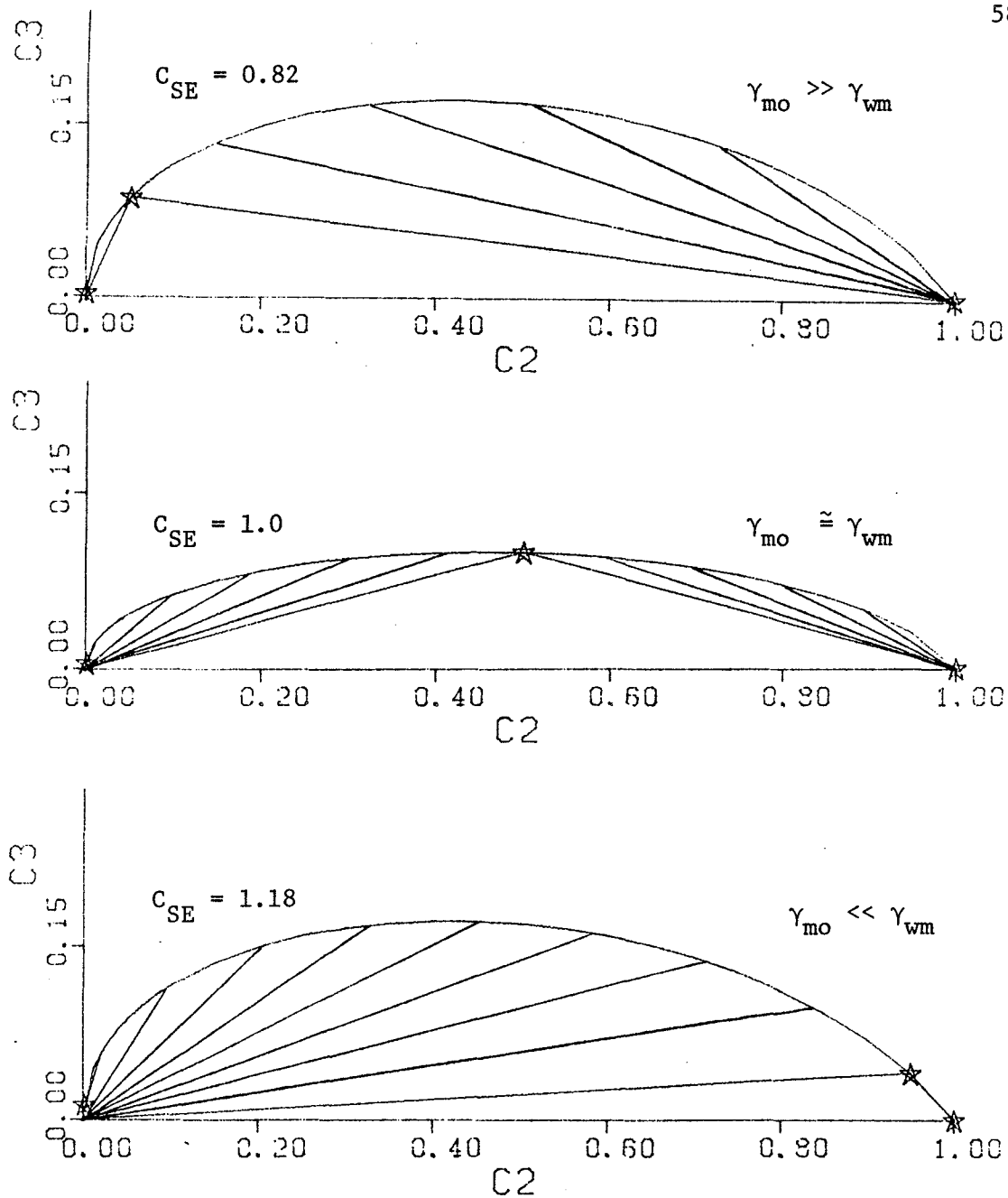


Figure 4.2. Phase diagrams and interfacial tensions in Type III phase behavior environment with different salinities.

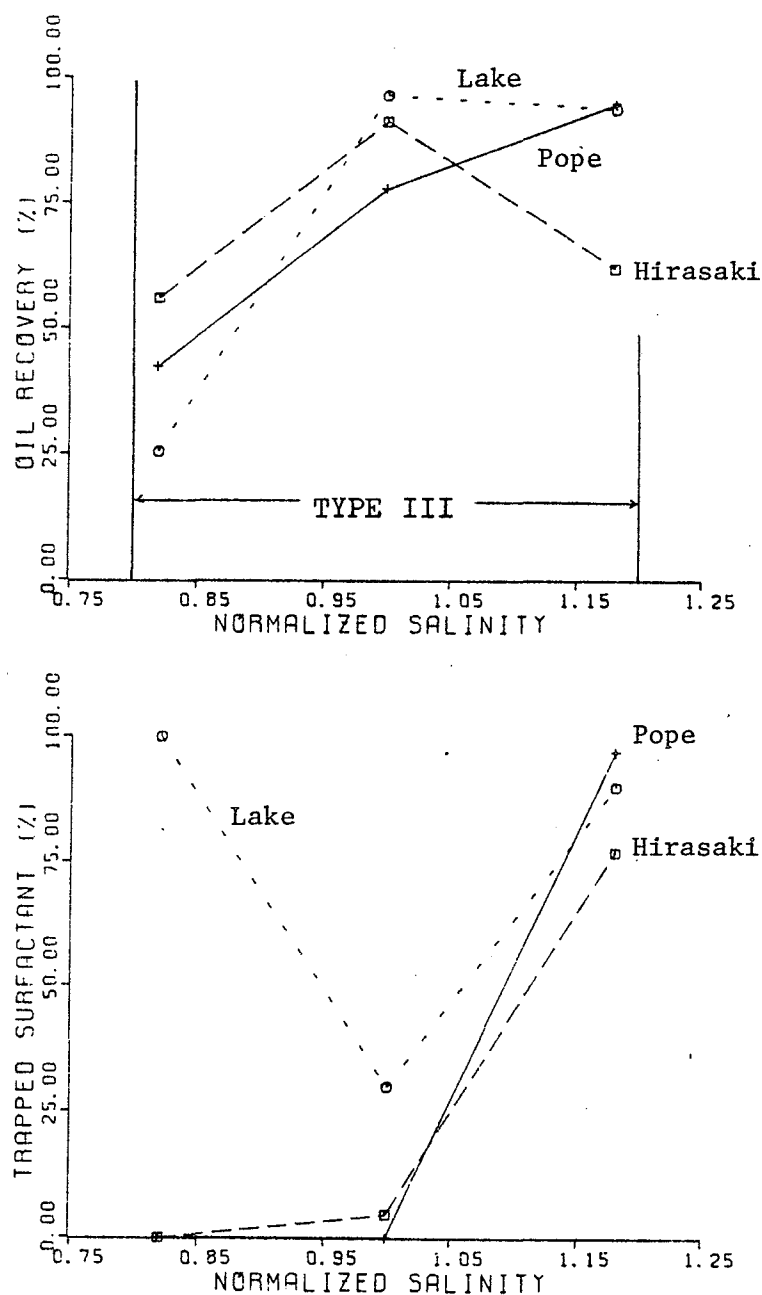


Figure 4.3. Comparisons of final oil recoveries and surfactant trappings among three different three phase flow models (0.1 PV of 3% surfactant slug is injected).

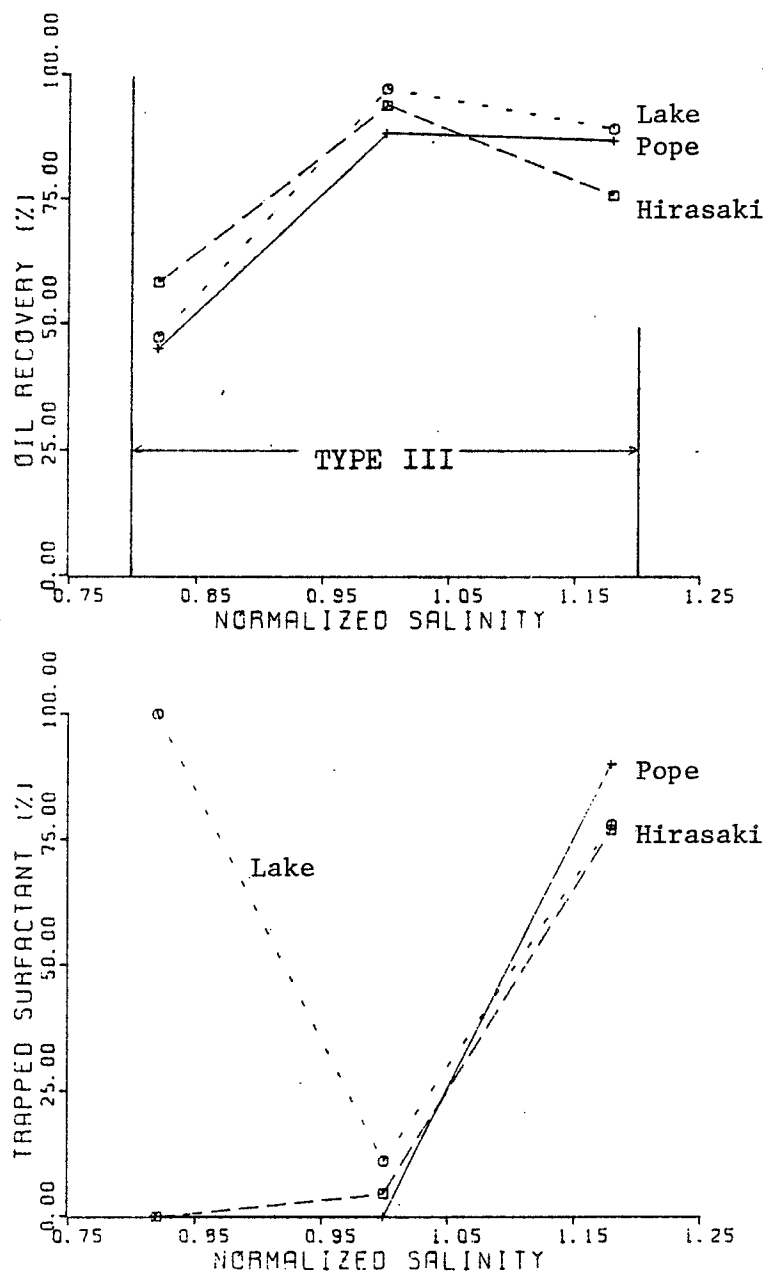


Figure 4.4. Comparisons of final oil recoveries and surfactant trappings among three different three phase flow models (0.1 PV of 6% surfactant slug is injected).

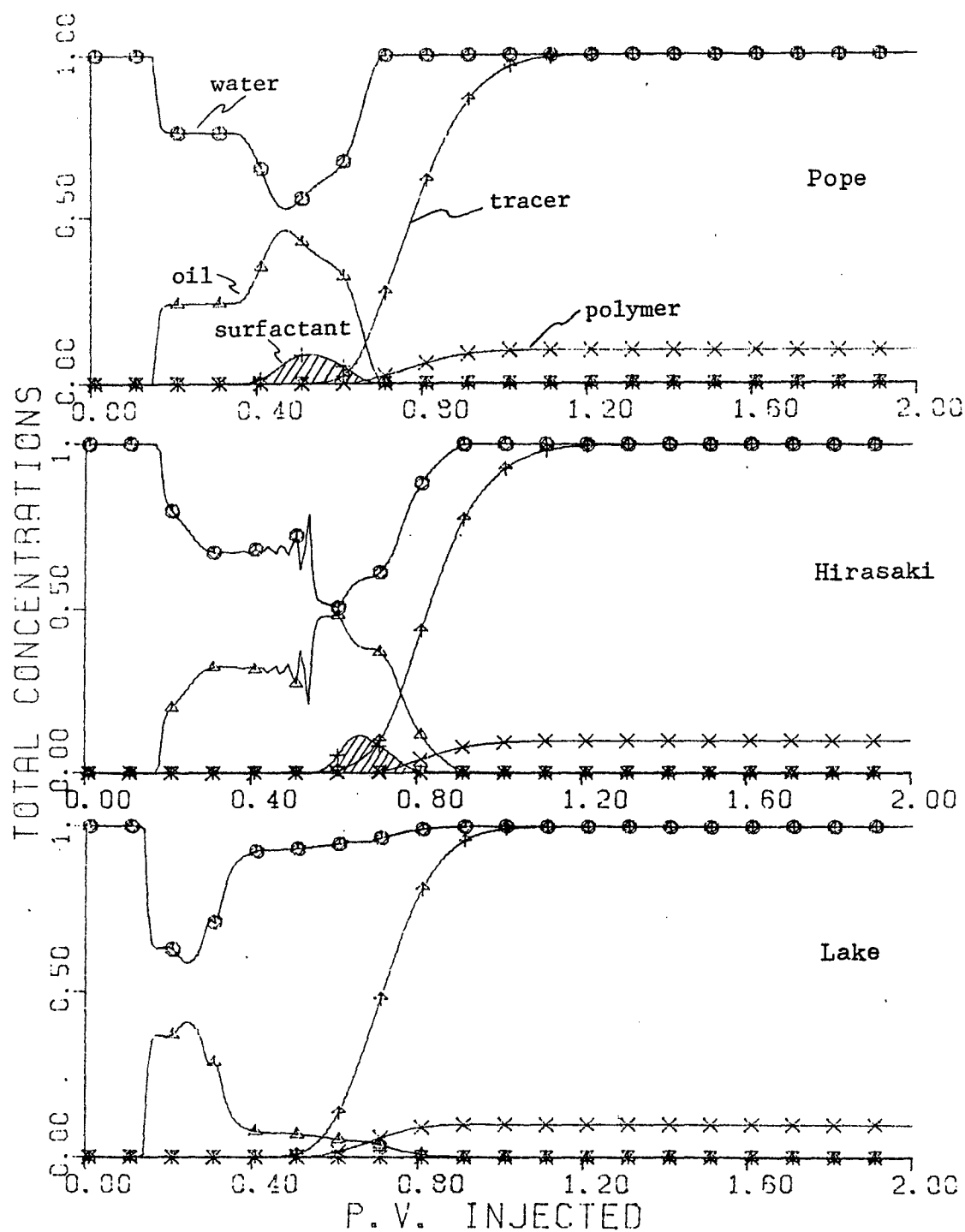


Figure 4.5. Comparison of production histories among three different three phase flow models ($C_{SE} \approx C_{SEL}$) (surfactant concentration is five times amplified).

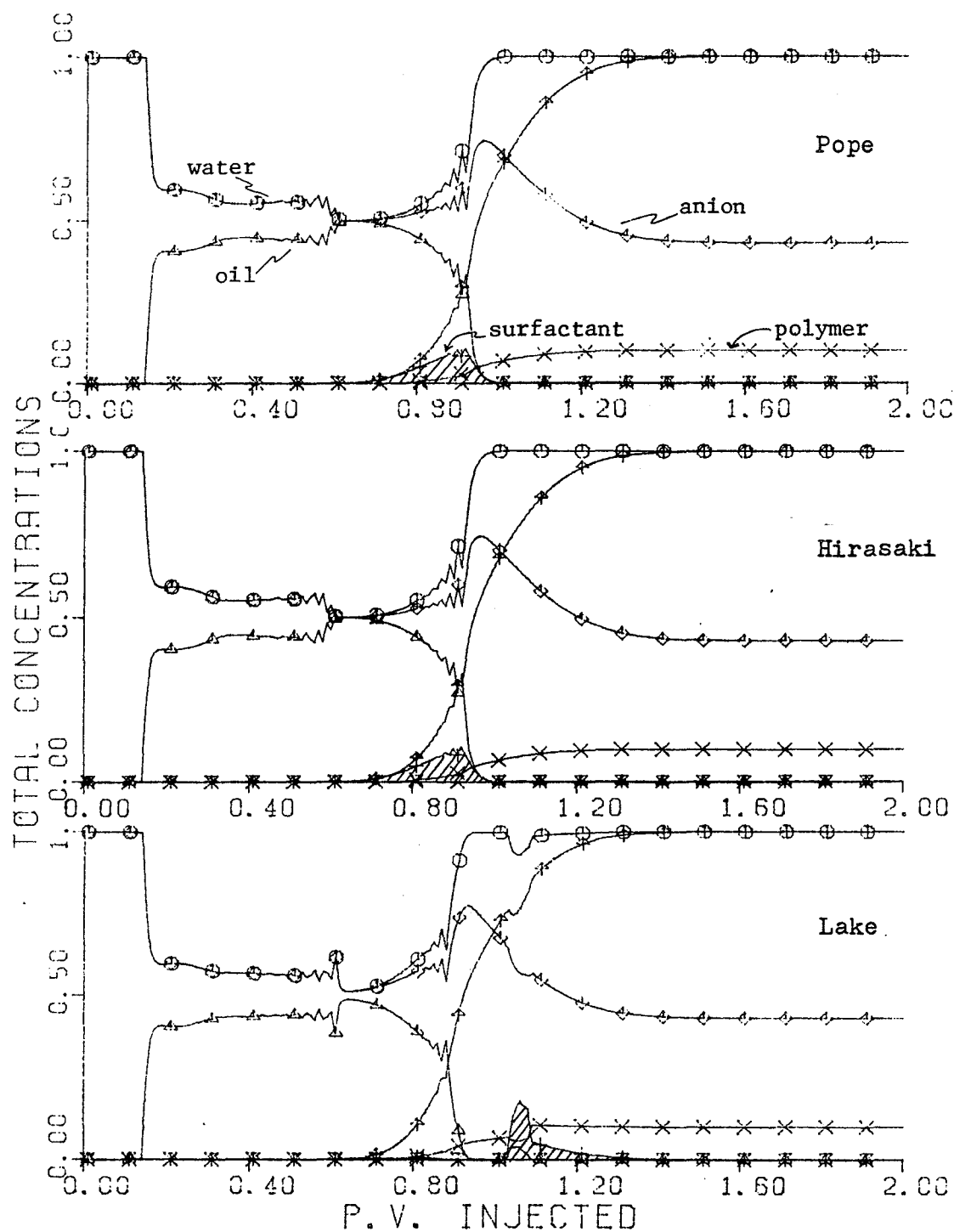


Figure 4.6. Comparison of production histories among different three phase flow models (salinity gradient) (surfactant concentration is five times amplified).

CHAPTER V

ORDINARY DIFFERENTIAL EQUATION INTEGRATORS

There exist quite a few numerical techniques to solve a system of first-order ordinary differential equations (ODE's) of the form

$$\frac{dy}{dt} \equiv y' = f(y,t) , \quad y(t_0) = y_0 \quad (5.1)$$

where y and f are vectors of length N . The techniques, in general, can be divided into two categories: single step methods and multistep methods.⁶¹

For single step methods, no information about the solution for previous steps is necessary. One representative example of such single step methods is the Runge-Kutta algorithm. Runge-Kutta methods require the evaluation of derivative $f(y,t)$ at intermediate points between the initial and end point of each step.

Multistep methods make use of information about the solution obtained from several previous steps to calculate the solution for the current step. Thus they generally require a larger amount of computer memory than the Runge-Kutta formulas of the same order. Concerning the computation, however, multistep methods can be rather economical integrators since they generally require only one or two functions evaluations per step.

5.1 Runge-Kutta-Fehlberg methods⁵⁶

Single step methods for solving $y' = f(y, t)$ require only a knowledge of the numerical solution y_n in order to compute the next value y_{n+1} . This has obvious advantages over the p -step multistep methods that use several past values $\{y_n, \dots, y_{n-p}\}$, and that require initial values $\{y_1, \dots, y_p\}$ that have to be calculated by another method.

The best known one-step methods are the Runge-Kutta methods; and they are the usual means for calculating the initial values $\{y_1, \dots, y_p\}$ for a $(p+1)$ -step multistep method. The major disadvantage of the Runge-Kutta methods is that they use many more evaluations of the derivative $f(y, t)$ to attain the same accuracy, compared with the multistep methods. At present, there are no variable order Runge-Kutta methods comparable to the Adams-Bashforth and Adams-Moulton methods. Runge-Kutta methods are closely related to the Taylor series expansion of $y(t)$, which is the solution of the initial value problem, but no differentiations of f is necessary in the use of the method.⁶⁰

The Fehlberg integrators are single-step, fixed-order methods and time-step size is varied according to the estimated truncation error made during the last time step. To estimate the truncation error, $(p+1)^{\text{th}}$ order and p^{th} order Runge-Kutta formulas are employed. The difference between those two approximations is defined to be an estimate of the leading term of the local truncation error for the p^{th} order approximation. Error is controlled by keeping the magnitude

of the local truncation error within some specified (desired) tolerance.⁶⁴ Depending on the algorithm, either the p^{th} or $(p+1)^{\text{th}}$ order approximation may be used as the solution.

A numerical solution of Equation (5.1) can be obtained by using either of the following Runge-Kutta-Fehlberg integration formulas

$$y_{i+1} = y_i + \Delta t \sum_{k=0}^{m-1} C_k f_k + O(\Delta t^{p+1}) \quad (5.2)$$

$$\hat{y}_{i+1} = y_i + \Delta t \sum_{k=0}^{\ell-1} \hat{C}_k f_k + O(\Delta t^{p+2}) \quad (5.3)$$

where m and ℓ are the number of function evaluations in the lower order and higher order formulas, respectively. Subscripts i and $i+1$ indicate their step level

$$f_0 = f(t_i, y_i) \quad (5.4)$$

$$f_k = f(t_i + \alpha_k \Delta t, y_i + \Delta t \sum_{\lambda=0}^{k-1} \beta_{k\lambda} f_\lambda) \quad (5.5)$$

where Δt is the step size. The constants α , β , c and \hat{c} are determined in the derivation of the algorithm. The superscript $\hat{}$ indicates higher order approximation.

An estimate of the local truncation error, LTE, for the lower-order solution is obtained as below

$$\text{LTE} \approx |y_{i+1} - \hat{y}_{i+1}| = \Delta t \left| \sum_{k=0}^{r-1} (C_k - \hat{C}_k) f_k \right| \quad (5.6)$$

where

$$r = \max(\ell, m)$$

Let

$$\text{TOL} = \text{ERR} * \max(|y|, \text{YBIAS}) \quad (5.7)$$

where ERR is the relative error tolerance. YBIAS is some specified lower limit to avoid the selection of a step size which is too small.

Here both LTE and TOL are a vector of length N, since y is a vector in our case. If any component of LTE is larger than the corresponding component of TOL, the step size is reduced and calculations are repeated until the desired accuracy is obtained. If every component of LTE is less than the corresponding component of TOL, the step size is accepted and the next step size to be used is calculated as follows

$$\Delta t_{\text{new}} = \text{PCT} * \Delta t_{\text{old}} * \min_{j=1, N} \left(\frac{\text{TOL}_j}{\text{LTE}_j} \right)^{1/(p+1)} \quad (5.8)$$

where PCT is a conservative factor, which is intended to prevent excessive step rejections. PCT may be assigned the value of 0.8 to 0.9. Δt_{old} is the current step size used. p is the order of approximation. N is the number of equations.

When a higher order ODE solver is applied to solve the continuity equation in the simulator, the truncation error produced from spatial discretization is much larger than the error from time integration. Consequently the pair of first and second order approximations

is considered to be sufficiently accurate and is used in our problem.

With the combination of first and second order approximation the constants in equations (5.2), (5.3), and (5.5) are given in Table 5.1.⁵⁷

Table 5.1 Coefficients for RK1(2).

k	α_k	β_{k0}	β_{k1}	c_k	\hat{c}_k
0	0	-	-	1/256	1/512
1	1/2	1/2	-	255/256	255/256
2	1	1/256	255/256	-	1/512

Although the number of function (derivative) evaluations is two and three for first and second order approximation, respectively, actually only two function evaluations are required per step. This is because the coefficients in Table 5.1 are determined with the intention of using the third evaluation again as the first evaluation for the next step. Since the combination of first and second order approximation is used and the former is taken to be the solution, this method is

called RK1(2) hereafter.

Substitution of the coefficients given in Table 5.1 into equations (5.2) through (5.5) yields

$$y_{i+1} = y_i + \Delta t \left(\frac{1}{256} f_0 + \frac{255}{256} f_1 \right) \quad (5.9)$$

$$\hat{y}_{i+1} = y_i + \Delta t \left(\frac{1}{512} f_0 + \frac{255}{256} f_1 + \frac{1}{512} f_2 \right) \quad (5.10)$$

$$f_0 = f(t_i, y_i) \quad (5.11)$$

$$f_1 = f\left(t_i + \frac{1}{2} \Delta t, y_i + \frac{1}{2} \Delta t \cdot f_0\right) \quad (5.12)$$

$$f_2 = f\left(t_i + \Delta t, y_i + \frac{1}{256} \Delta t \cdot f_0 + \frac{255}{256} \Delta t \cdot f_1\right) \quad (5.13)$$

Then an estimate of local truncation error is obtained by subtracting equation (5.10) from equation (5.9)

$$|LTE| \cong \left| \frac{\Delta t}{512} (f_0 - f_2) \right| \quad (5.14)$$

Since lower order approximation (5.9) is taken as the solution rather than higher order approximation (5.10), the approximation is first order. Then equation (5.8) is now

$$\Delta t_{\text{new}} = \text{PCT} * \Delta t_{\text{old}} * \min_{j=1, N} \left(\frac{\text{TOL}_j}{\text{LTE}_j} \right)^{1/2} \quad (5.15)$$

From Taylor series expansions, equation (5.15) below can be easily derived

$$y_{i+1} = y_i + \Delta t \left(\frac{1}{256} \frac{dy}{dt} \Big|_i + \frac{255}{256} \frac{dy}{dt} \Big|_{i+\frac{1}{2}} \right) + \frac{\Delta t^2}{512} \frac{d^2y}{dt^2} \Big|_i + o(\Delta t^3) \quad (5.16)$$

which suggests the truncation error of approximation (5.9) used to calculate y_{i+1} is

$$|LTE| \cong \left| \frac{\Delta t^2}{512} \frac{d^2y}{dt^2} \Big|_i \right| \quad (5.17)$$

This is smaller than the one for Euler's method by a factor of two hundred and fifty-six.

The calculation procedure is as follows:

- 1) Given y_i , t , and Δt
- 2) Calculate f_0 from equation (5.11)
- 3) Calculate f_1 from equation (5.12)
- 4) Calculate y_{i+1} from equation (5.9)
- 5) Calculate f_2 from equation (5.13)
- 6) Calculate LTE from equation (5.14)
- 7) If LTE is less than TOL, time step size is accepted and new time step size is calculated from equation (5.15).
Then resetting $f_0 = f_2$, $y_i = y_{i+1}$ and $t = t + \Delta t_{old}$, go to step (3) and start calculation of next step.
- 8) Otherwise time step size Δt is reduced and repeat the calculation from step (3).

5.2 Multistep Methods

The class of linear multistep methods for ODE's integration is usually described as follows: Approximate solution values are calculated at $t = t_0, t_1, t_2, \dots$ where $t_n = t_{n-1} + \Delta t$, with step size Δt , according to a formula of the form

$$y_n = \sum_{j=1}^{k_1} \alpha_j y_{n-j} + \Delta t \sum_{j=0}^{k_2} \beta_j y'_{n-j} \quad (5.18)$$

where $y_k = y(t_k)$, $y'_k = y'(t_k) = f(y_k, t_k)$, α_j and β_j are coefficients associated with the particular method. Equation (5.18) is used to calculate y_n when previous approximate values of y and y' are known. Special considerations have to be made to obtain the several values needed at the beginning to make equation (5.18) applicable. The most popular examples of equation (5.18) fall into two specific classes of methods. One is referred to as "the Adams' methods of order q " and is obtained by setting $\alpha_1 = 1$, $k_1 = 1$ and $k_2 = q - 1$. The other is the backward differentiation methods (usually called Gear's methods) of order q which is obtained by setting $k_1 = q$ and $k_2 = 0$. When order is said to be q , it means that if Eq. (5.18) is solved for y_n with all past values being exact, then y_n will differ from the correct solution of Eq. (5.1) by a truncation error that is $O(\Delta t^{q+1})$ for small Δt .

The biggest advantage of an Adams' integrator over Gear's methods is that it does not require the evaluation of the Jacobian, nor solving a large matrix problem in its solution process, since a fixed

point iteration is used to solve the non-linear (corrector) equations. The main disadvantage, on the other hand, is that their stability regions are small and can often require relatively small time steps to maintain stability. This disadvantage makes these methods inefficient for stiff problems.

Gear's methods of order $1 \leq q \leq 6$ were shown to have stiff stability by Gear.⁵⁸ Their stability region contains a horizontal strip covering the entire negative real axis in all six cases (Figure 5.3). The boundary curve crosses the axis, making the method not stiffly stable for $q \geq 7$. Newton's method rather than a fixed point iteration is used to solve corrector equations. A fixed point iteration imposes time step size limitations to make corrector converge, which destroys the advantage gained by achieving stiff stability. The expense of calculating the Jacobian matrix $\partial f / \partial y$ can be further offset by neglecting to re-evaluate it at every step, unless the existing value of this matrix fails to produce convergence or the order q is changed.

Sophisticated and highly reliable computer programs have been developed for solving complicated systems of ODE's, using either Adams' methods or Gear's stiffly stable methods. Allowing users to specify which methods be used, these programs automatically select the order q and time step size keeping the error produced from the integration within the desired tolerance and maintaining the time step size as large as possible. In this research, one such program named DGEAR⁵⁹ is applied to micellar/polymer flooding simulation. The code DGEAR implements the

Adams' methods of order $1 \leq q \leq 12$ and Gear's method of order $1 \leq q \leq 5$.

With DGEAR, the user may choose from several different algorithms only by specifying two method indicators. The first is called METH, which indicates the method of integration to be used. The second is called MITER, which indicates the procedure for solving the nonlinear equations arising in the method being used. The description of those parameters is given below.

METH = 1, indicates Adams' method

2, indicates backward differentiation (Gear's) method

MITER = 0, implies functional (or fixed point) iteration.

The Jacobian is not needed.

1, implies a chord method (or semistationary Newton iteration) with the Jacobian supplied by the user.

2, implies a chord method with the Jacobian calculated internally by finite differences.

3, implies a chord method with the Jacobian replaced by a diagonal approximation based on a directional derivative.

5.3 Stability region and stiffness

When an ordinary differential equation is integrated numerically, the stability of the method is often discussed, because it suggests the quality of the solution or step size required to obtain an accurate solution. To investigate the stability, only the special equation below and its region of absolute stability are usually considered.

$$y' = \lambda y \quad (5.19)$$

Here λ is a complex constant having a negative real part. The region of absolute stability is defined⁶⁰ as the set of all $\lambda\Delta t$ (Δt : step size, therefore real non-negative) for which the numerical solution $y_n \rightarrow 0$ as $t_n \rightarrow \infty$. The larger the region of absolute stability, the less the restriction on Δt in order to have the numerical method give a numerical solution that is qualitatively the same as the true solution. If a linear system of ordinary differential equations $y' = Ay$ is being considered, the eigenvalues λ_i ($i = 1, 2, \dots, N$) of matrix A , instead of λ in equation (5.19), are used to determine the step size which gives a stable solution. Furthermore, the stability of the methods used in solving a system of nonlinear equations can be determined by considering the eigenvalues of the Jacobian matrix, $\partial y' / \partial y$.

When a p^{th} order Runge-Kutta scheme with r function evaluations is applied to the test equation (5.19), an approximate solution of the form⁶¹

$$y_{i+1} = \{\pi_r^p(\lambda\Delta t)\}y_i \quad (5.20)$$

will be obtained, where

$$\pi_r^p(\lambda\Delta t) = \sum_{j=0}^p \frac{(\lambda\Delta t)^j}{j!} + \sum_{q=p+1}^r \gamma_q (\lambda\Delta t)^q \quad (5.21)$$

the γ_q , $q = p+1, p+2, \dots r$ are functions of α, β, c (the coefficients of the Runge-Kutta formula used in integration of the test equation). See equation (5.2). $\pi_r^p(\lambda\Delta t)$ is called the stability polynomial. The region of absolute stability is the area defined by

$$|\pi_r^p(\lambda\Delta t)| < 1 \quad (5.22)$$

This ensures that the error does not increase from step to step in the numerical solution of the test problem. One way of defining a stability region is to get

$$\pi_r^p(\lambda\Delta t) = e^{i\theta}$$

or

$$\pi_r^p(\lambda\Delta t) = \cos\theta + i\sin\theta \quad i = \sqrt{-1} \quad (5.23)$$

By varying $0^\circ < \theta < 360^\circ$ and calculating the roots of equation (5.23) for the small variation of θ , one can obtain the boundaries of the absolute region of stability.

Now let us consider equation (5.9) of RK1(2)

$$y_{i+1} = y_i + \Delta t \left(\frac{1}{256} f_0 + \frac{255}{256} f_1 \right) \quad (5.9)$$

Substituting the test equation $f = \lambda y$ into equations (5.11) and

(5.12)

$$\begin{aligned}
 f_0 &= f(t_i, y_i) \\
 &= \lambda y_i
 \end{aligned}
 \tag{5.24}$$

$$\begin{aligned}
 f_1 &= f(t_i + \frac{1}{2} \Delta t, y_i + \frac{1}{2} \Delta t f_0) \\
 &= \lambda(y_i + \frac{1}{2} \Delta t \lambda y_i)
 \end{aligned}
 \tag{5.25}$$

Then substitution of equations (5.24) and (5.25) into equation (5.9) gives

$$y_{i+1} = (1 + \lambda \Delta t + \frac{255}{512} \lambda^2 \Delta t^2) y_i \tag{5.26}$$

This result agrees with equations (5.20) and (5.21) as below

$$\pi_r^p(\lambda \Delta t) = \pi_2^1(\lambda \Delta t) = \sum_{j=0}^1 \frac{(\lambda \Delta t)^j}{j!} + \sum_{q=2}^2 \gamma_q (\lambda \Delta t)^q \tag{5.27}$$

where

$$\gamma_2 = \frac{255}{512}$$

To obtain absolute stability region, let

$$1 + \lambda \Delta t + \frac{255}{512} \lambda^2 \Delta t^2 = \cos \theta + i \sin \theta \tag{5.28}$$

solving equation (5.28) for $\lambda\Delta t$ with varying θ , the absolute stability region is given in Figure 5.1. Since the stability region is symmetric, only the upper half is shown in the figure.

When a multistep method expressed by equation (5.18) is applied to test equation (5.19), we obtain⁶⁰

$$y_{i+1} = \sum_{j=0}^p a_j y_{i-j} + \lambda\Delta t \sum_{j=-1}^p b_j y_{i-j}$$

$$(1 - \lambda\Delta t b_{-1})y_{i+1} - \sum_{j=0}^p (a_j + \lambda\Delta t b_j)y_{i-j} = 0 \quad i \geq p \quad (5.29)$$

This is a homogeneous linear difference equation of order $p+1$, and the theory for its solvability is completely analogous to that of $(p+1)$ st order homogeneous linear differential equations. We attempt to find a general solution by first looking for solutions of the special form

$$y_i = r^i \quad i \geq 0 \quad (5.30)$$

If we can find $p+1$ linearly independent solutions, then an arbitrary linear combination will give the general solution of (5.28).

Substituting $y_i = r^i$ into (5.29) and cancelling r^{i-p} , we obtain

$$(1 - \lambda\Delta t b_{-1})r^{p+1} - \sum_{j=0}^p (a_j + \lambda\Delta t b_j)r^{p-j} = 0 \quad (5.31)$$

This is called the characteristic equation and the lefthand side is the characteristic polynomial. The roots are called characteristic roots. If the roots are all distinct, then the general solution of equation (5.28) is

$$y_i = \sum_{j=0}^p \gamma_j [r_j(\lambda \Delta t)]^i \quad i \geq 0 \quad (5.32)$$

where $r_j(\lambda \Delta t)$ are characteristic roots, which depend continuously on the value of $\lambda \Delta t$. If $r_j(\lambda \Delta t)$ is a root of multiplicity $n > 1$, then the following are n linearly independent solutions of (5.29).

$$[r_j(\lambda \Delta t)]^i, i[r_j(\lambda \Delta t)]^i, \dots, i^{n-1}[r_j(\lambda \Delta t)]^i \quad (5.33)$$

These can be used with the solutions arising from the other roots to generate a general solution for (5.29), comparable to equation (5.32).

From equation (5.32), the region of absolute stability is equivalent to the area that satisfies

$$r_j(\lambda \Delta t) \leq 1 \quad 0 \leq j \leq p \quad (5.34)$$

The regions of absolute stability for Adams' methods and Gear's methods are shown in Figures 5.2 and 5.3 respectively.

If a system of linear or nonlinear ordinary differential equations is under consideration, all eigenvalues λ_i ($i = 1, 2, \dots, N$) associated with the matrix must satisfy the absolute stability condition.

Consequently one difficulty may arise when $\max_{i=1,N} |\lambda_i|$ is much larger than $\min_{i=1,N} |\lambda_i|$. In such a case, a limited stability region may impose a severe restriction on the step size. Such a system is called stiff and the degree of stiffness can be expressed by stiffness ratio SR

$$SR = \frac{\max_{i=1,N} |\lambda_i|}{\min_{i=1,N} |\lambda_i|} \quad (5.35)$$

If a system is stiff, it involves both rapidly changing variables and very slowly changing variables, all of a decaying nature.

Let us consider more specific problems such as semi-discrete systems arisen from convection-diffusion equations. The more parabolic (diffusive) the character of the equation is, the higher the degree of stiffness. In general, a larger number of grid blocks yields a stiffer system.

Backward differentiation methods (Gear's methods) are especially designed for stiff problems. As can be seen in Figure 5.3, they have infinite regions of absolute stability and offer higher efficiencies to solve stiff problems.

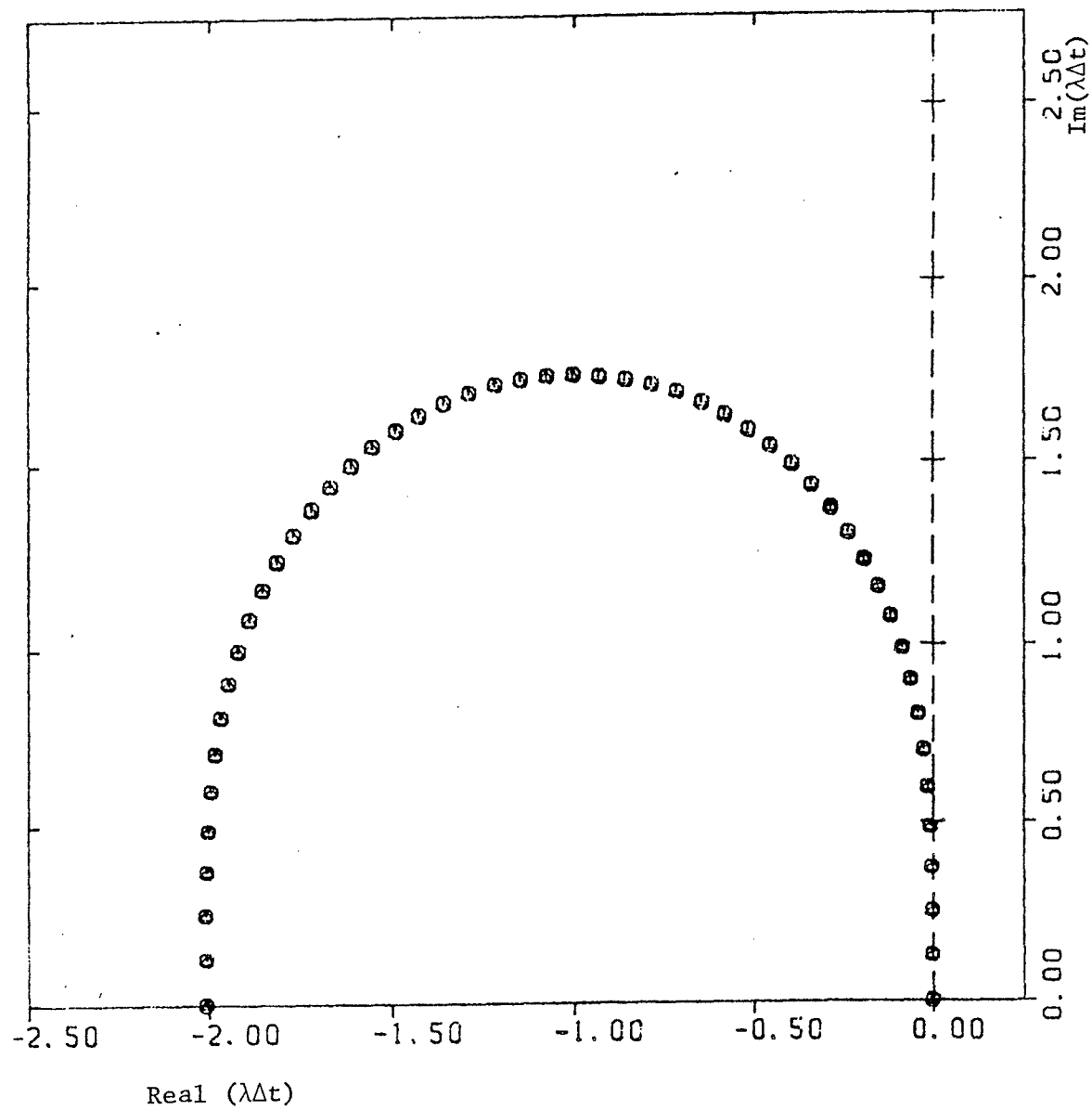


Figure 5.1. Stability region for RK1(2).

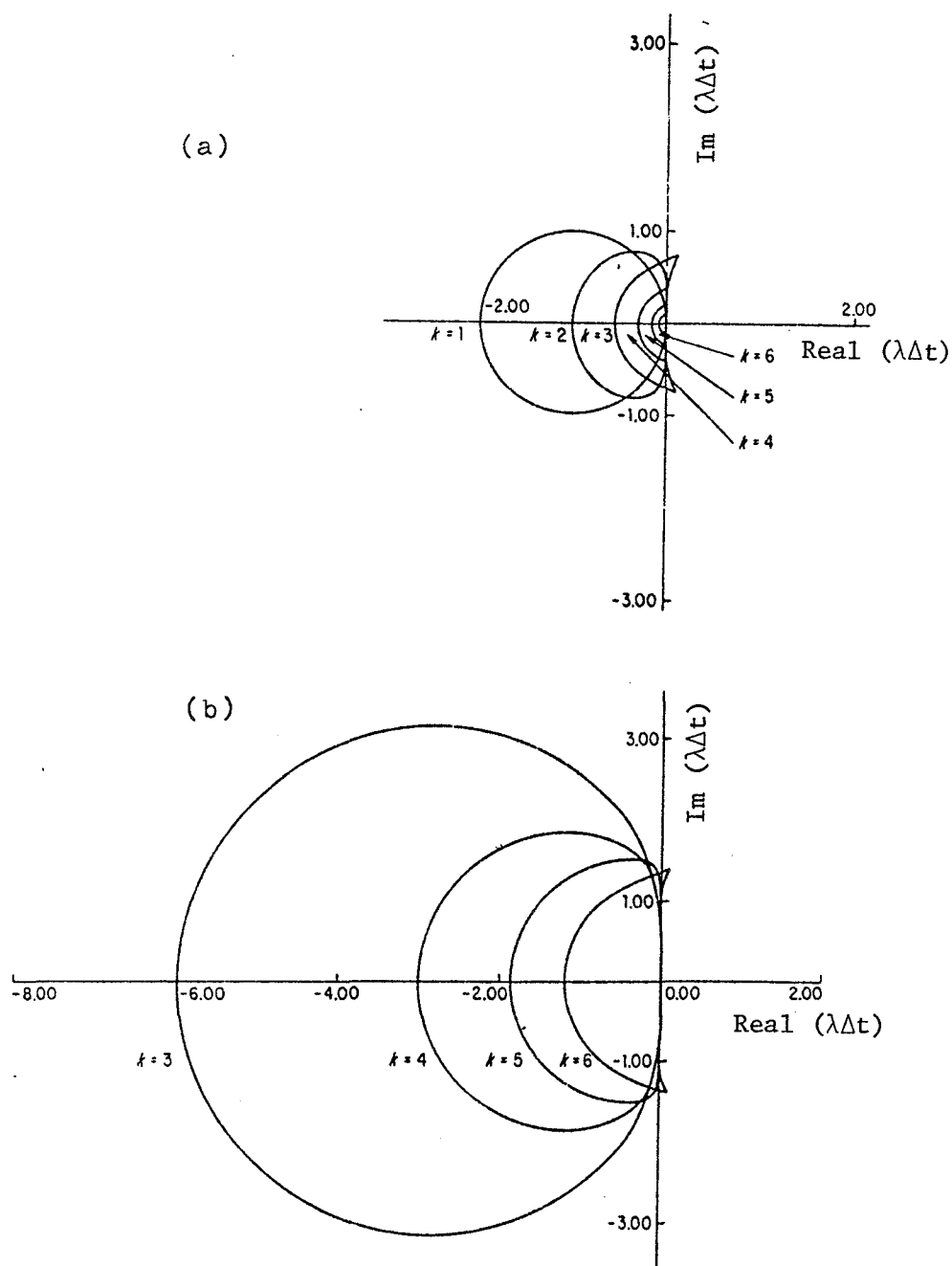


Figure 5.2. Stability region for Adams' methods.⁵⁸ Method of order k is stable inside region indicated.

(a) Adams-Bashforth methods.

(b) Adams-Moulton methods.

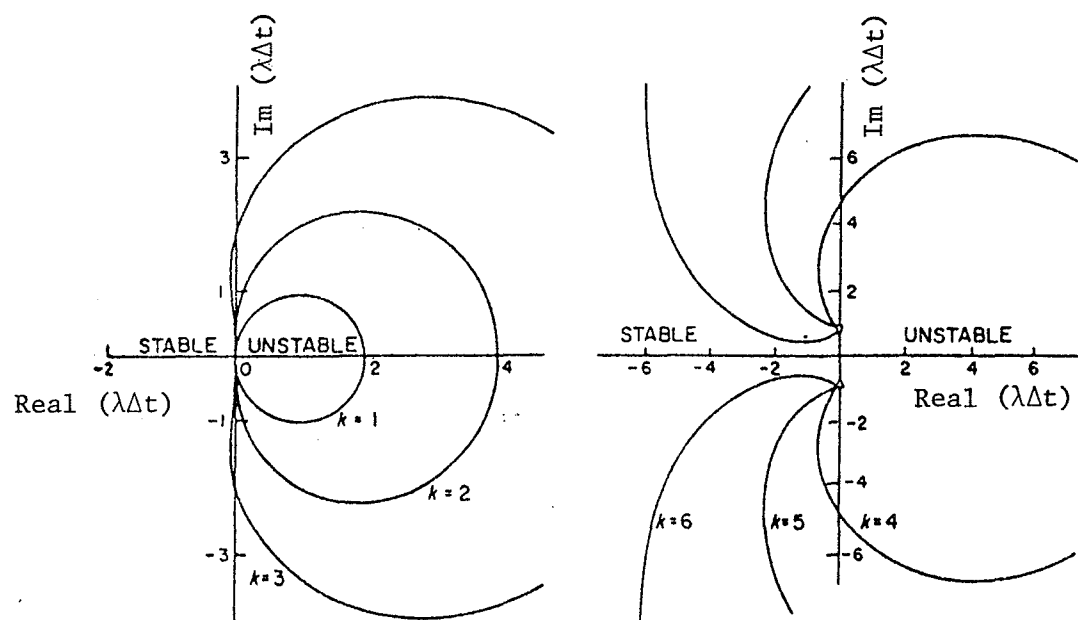


Figure 5.3. Stability region for Gear's methods.⁵⁸
 (k-step backward differentiation method)

CHAPTER VI

RESULTS AND DISCUSSIONS

In this chapter, the basic input data and a set of computed results for a type II(-) phase environment with no adsorption are first presented. The computed results were obtained from the fully-discrete Euler (FDE) method with constant time step size of 0.001 Pore Volumes (PV). These plotted results enable us to understand how the flood is proceeding.

In the next section, results of each method for Type II(-) phase environment with no adsorption are first presented with some discussion. Although RK1(2) was the best among semi-discrete methods, computation time was not improved as had been expected. Then an alternative algorithm which is called RK1 is introduced and its results are compared with FDE method.

The effect of adsorption is shown in Section 6.3 for only RK1 and RK1(2). More computation time was required for both RK1 and RK1(2) when adsorption is involved compared with the no adsorption cases. Other phase behavior environments were also examined in this section. The performance of the semi-discrete method got worse as the phase behavior environment was changed from Type II(-) to Type II(+), and was worst in the Type III phase behavior environment.

For the Type II(-) phase behavior environment with no adsorption, the effect of each component on the time step size selection

was examined (in Section 6.4) using the RK1 method. It was found that RK1 was taking a too conservative (small) time step size at an early stage of the flood. This is due to rather large truncation error associated with the time integration at that stage for the components which did not exist before the micellar flooding started. When the error was checked only for the water or oil component, the computation time was improved without affecting the quality of the solution.

In section 6.5, the results for the case when the absolute error was specified as the error tolerance instead of the relative error. By using a smaller value for PCT in equation (5.8), less computation time was achieved with absolute error control. A brief summary of RK1 for Type II(-) without adsorption is also given.

Some eigenvalues of the Jacobian matrix before surfactant breakthrough are presented in section 6.6. Since the matrix is characterized as block triangular, it is possible to decouple it into submatrices and obtain eigenvalues by analyzing each submatrix. The eigenvalues presented are obtained from the oil bank blocks which seem to be governing the stability requirement.

A stability analysis with some assumptions was performed (in section 6.7) for the blocks where surfactant is present. From this analysis, it was discovered that an unconditional instability occurred continually and locally due to the change in residual saturation caused by the surfactant. This instability explains why we have oscillations in the production history even if a small time

step is used.

6.1 Basic input data and example results

In this section the basic input data used to compare the performance of the various semi-discrete methods discussed so far are presented. Then a set of computed results for Type II(-) phase environment with no adsorption are shown with some discussion. The computed results were obtained from the FDE method with constant time step size of 0.001 PV. These plotted results enable us to understand how the flood is proceeding.

The basic input data are listed in Tables 6.1. Tables 6.1a and 6.1b list input data common for all base cases while Table 6.1c lists the difference in the data for each base case. A definition of each parameter is given in Appendix C. Some representative features of the input data will be discussed later in this section. Unless otherwise noted, this basic data set is used for all runs. A somewhat simplified case is considered to enable easier interpretation of the calculated results. The main purpose of this research is to investigate the applicability of the semi-discrete methods to micellar/polymer simulation, and to compare the performance among different semi-discrete methods. For $RK1(2)$ PCT in equation (5.15) and YBIAS in equation (5.7) are fixed to be 0.8 and 10^{-3} , respectively.

The common features for all base cases in Tables 6.1 are as follows:

- 1) The physical dispersion term is not used assuming that numerical dispersion can adequately approximate physical dispersion.
- 2) The dimensionless longitudinal dispersivity given as α_{Dj} in equation (3.1) is about 0.0125 since the number of grid blocks used is forty.
- 3) Alcohol is combined with chemical to make a single pseudo component.
- 4) No ion exchange between clay and mobile phases is considered.
- 5) No inaccessible pore volume to polymer or surfactant is considered.
- 6) The initial condition is waterflood residual oil saturation.
- 7) No salinity effect or shear rate effect on polymer solution viscosity is included. Permeability reduction due to polymer is not considered either.
- 8) The plait point is located at the corner of the pseudo ternary diagram, which yields excess phases consisting of a single pure component (either water or oil).
- 9) An aqueous surfactant slug containing 10% (surfactant + alcohol) and 0.1 wt % polymer is first injected up to 0.1 pore volume (PV). Then polymer buffer of constant polymer concentration equal to 0.1 wt % is injected.
- 10) Calculated viscosities of surfactant slug and polymer

buffer are 33 cp and 31 cp, respectively.

- 11) Tracer is injected in both surfactant and polymer slug.
- 12) Total injected amount is 2.0 PV.

Figures 6.1 through 6.10 present a set of computed results for a Type II(-) phase behavior environment with no adsorption (Run 207). The FDE method with a time step size of 0.001 was used. Figures 6.1 through 6.7 show profiles at 0.25 PV injected. Figures 6.8 to 6.10 show the production histories.

Since no adsorption was considered and only microemulsion and excess oil exist, the fronts of surfactant, polymer, and tracer are located at the same position. There are two major fronts in the profiles. One is at the surfactant front and the other is at the most upstream grid. The former advances as injection proceeds, while the latter stays at the same place in this example. At the surfactant front, interfacial tension is reduced, which leads to the reduction of non-wetting phase residual saturation. At the most upstream grid, oil saturation is extremely low, which is considered to be the effect of a miscible displacement. However, such low residual oil saturation is not achieved in this example beyond that grid because of the dilution of the surfactant.

In Figures 6.2 through 6.4 vertical lines indicate the appearance or disappearance of surfactant. When the total concentration of (surfactant + alcohol) is lower than 10^{-4} , the effect of surfactant is neglected and no microemulsion phase is considered. Although the vertical lines separate water phase and microemulsion

phase, these two phases are considered to be continuous in Type II(-) case.

Figure 6.7 shows the profile of total relative mobility. This figure indicates that physically unstable condition occurs at the surfactant front. This is caused by the increase in relative permeabilities both to oil and aqueous (microemulsion) phases due to surfactant. However, if the mobilities are compared between oil bank and the middle of surfactant slug, the mobility ratio is about 0.4. Mobility ratio for surfactant slug and polymer buffer is designed to be about unity. Thus the whole process is stable even though locally unstable.

6.2 Comparison of each method for Type II(-) phase behavior with no adsorption

In this section a comparison is made among the fully-discrete Euler (FDE) method and the semi-discrete methods discussed so far. Type II(-) phase behavior environment with no adsorption (data set A in Table 6.1c) was adopted as the simplest example. The results are shown in Table 6.2 and in Figures 6.11 through 6.19. The legend of total concentration history plots is given in Table 6.15. This legend is valid for all subsequent total concentration plots. Although RK1(2) was the best among semi-discrete methods, computation time was not improved as had been expected. Then an alternative algorithm which seemed to be more efficient is introduced. This algorithm called RK1 consists of a combination of first

and second order Runge-Kutta approximations. Its results are also compared with the FDE method.

In Table 6.2, ERR is the relative error tolerance for semi-discrete methods while constant time step size used is written for FDE in the same column. IEVA is the number of function (derivative) evaluations. NREJ indicates how many times the predicted Δt_D has been rejected. NREJ was counted only for RK1(2). CPU time, which is for CDC Dual Cyber 170/750 at the University of Texas at Austin is listed just to give an idea of the order of the computer time. One should compare IEVA rather than CPU time to see the efficiency of each method, since the semi-discrete methods were programmed mainly to see their applicability and flexibility. There may be more room to decrease CPU time for semi-discrete runs. In the column of quality in Table 6.2, "good" means the solution obtained looks comparable to Run 207, which used FDE method with $\Delta t_D = 0.001$. Fair quality means the solution oscillates somewhat but is still acceptable. In last column, E_R is the total oil recovery as percent of the initial oil in place. The oil recoveries are presented to show the effect of the change in numerical dispersion.

Figures 6.11 to 6.13 show the history of total concentration in the effluent for the FDE method with three different time step sizes. In Figures 6.14 through 6.16, total concentration histories are presented at the top and the histories of the time step size used at the bottom for the RK1(2) method, with varying ERR. Figures 6.17 and 6.19 are the same except for Adams' methods. For Adams'

methods, only the results with the option $\text{MITER} = 0$ are presented because other options did not give good results. The results of backward differentiation are also excluded because of their poor performance. Some discussions on these predictor-corrector methods will be presented later in this section.

From Table 6.2 one may conclude that the FDE method is better than the semi-discrete methods tested based on the number of function evaluations and CPU time. Actually, RK1(2) is not so efficient as was expected. Neither are Adams' and Gear's methods. However, one should keep in mind that with these methods the truncation error associated with time integration is controlled and forced to be smaller than some specified value, which was not done with the FDE method. This feature may become important when higher order accurate approximations for space derivatives or finite element methods are introduced. Furthermore, the history of the time step size should not necessarily be constant. Before surfactant breakthrough, the time step size taken with RK(2) is 0.002 to 0.003 PV, which coincides with the result obtained from the FDE method. When the FDE method is used, a time step size larger than 0.003 PV produces oscillation, which seems to be caused by numerical instability. After surfactant breakthrough, however, the time step size more than doubles with RK1(2) and finally increases to DTMAX. DTMAX is the maximum time step size specified in the input data. After a time step size is calculated based on the estimated truncation error, the time step size is compared with DTMAX and the smaller

value is taken to be the next time step size.

In Table 6.2, an extremely small error tolerance was listed for RK1(2). This is because RK1(2) produces a very small truncation error. When a larger error tolerance was used, RK1(2) selected larger time step sizes due to the small truncation error, and the solution was no longer stable. Oscillations in both concentration and time step size history occurred.

Although the time step size was increased after surfactant breakthrough, RK1(2) still required more computation time than the FDE method. This is because RK1(2) requires two function evaluations (equations (5.12) and (5.13)) per step. This means that the average time step size taken with RK1(2) has to be more than twice as large as the one for the FDE method. For this reason, another method called RK1 which requires only one function evaluation per step was tested. This method discussed below also has time step size control in a similar way to RK1(2).

Instead of equations (5.9) through (5.13) in Chapter 5, equations (6.1) through (6.4) below are used to estimate the local truncation error.

$$y_{i+1} = y_i + \Delta t f_0 \quad (6.1)$$

$$\hat{y}_{i+1} = y_i + \Delta t \left(\frac{1}{2} f_0 + \frac{1}{2} f_1 \right) \quad (6.2)$$

$$f_0 = f(t_i, y_i) \quad (6.3)$$

$$f_1 = f(t_i + \Delta t, y_i + \Delta t f_0) \quad (6.4)$$

Approximation (6.1) is exactly the same as Euler's method, while approximation (6.2) is the trapezoidal method solved with one iteration using Euler's method as the predictor. Both are a special case of Runge-Kutta formulas. Subtracting equation (6.2) from equation (6.1), the local truncation error LTE is estimated as below

$$|LTE| = \left| \frac{1}{2}(f_0 - f_1) \right| \quad (6.5)$$

Since Euler's formula is used as the solution, the truncation error is two hundred and fifty six times more compared with RK1(2)

$$|LTE| = \left| \frac{\Delta t^2}{2} \frac{d^2 y}{dt^2} \right|_1 \quad (6.6)$$

Its stability region is also the same as the one for the forward Euler method, which is shown in Figure 6.39.

Some results obtained using RK1 are presented in Table 6.3 where the results of FDE method are again shown for comparison. Plotted histories are presented in Figures 6.20 through 6.22. Those results show that RK1 saves computation time of about 20% compared with FDE method for the same quality. Furthermore the number of function evaluations is about 40% less for RK1 than the FDE method.

None of the predictor-corrector methods, neither Adams' methods nor Gear's method worked as well as RK1(2) or RK1. Although

several options were tested with DGEAR, only the results of Adams' methods with MITER = 0, which implies the functional (fixed point) iteration, are presented in Table 6.2, because all the other options did not work as well.

When Adams' methods with MITER = 0 were used, the order was varied from first to third. Although the average number of function evaluations per step was about two, only one function evaluation was the most frequent result. This fact indicates that the stability regions of Adams' predictor method play an important role in the time step size selection. Since the stability regions of second and third order Adams' predictors (Figure 5.2a) are smaller than the one for RK1(2) (Figure 5.1) or RK1 (Figure 6.39), Adams' methods may have to take a smaller time step than RK1(2) or RK1 to remain stable.

MITER = 0 was not tested with Gear's methods, since this iteration scheme imposes a limitation on the time step size which destroys the advantage of the stiffly stable character of Gear's methods.

Table 6.4 shows the results obtained with MITER equal to two. This option uses semi-stationary Newton iteration with the Jacobian calculated internally by finite differences. When the MITER = 2 option was tested, the number of grid blocks was decreased to twenty, because this option requires a large storage for the Jacobian matrix. In Table 6.4, ERR designates the relative error tolerance and Δt_D is the dimensionless time step size. Highest order q means the highest order of the methods used in the computation.

IEVA is the number of function evaluations including the evaluation of the Jacobian. NSTEP is the number of steps. NJE is the number of Jacobian evaluations. Net IEVA is the number of function evaluations excluding the number of function evaluations used to obtain the Jacobian matrix. Since 120 function evaluations were required to get a Jacobian matrix, most of the computation time was spent to evaluate Jacobian matrices.

The results were obtained only up to 0.1 PV injection because the solution oscillates, or large error occurred, for all runs after 0.1 PV injection. This is the time at which the composition of the injected slug is changed.

Although neither Adams' methods nor Gear's methods worked as well as the two explicit techniques, it may be too early to conclude these implicit or semi-implicit methods are not as good. Since a packaged program was used to implement the techniques, the details are not clear, but there may be some improvement possible. One example is the step size control. DGEAR varies time step size based on only relative error without YBIAS which was used for the two explicit methods as in equation (5.7). Since most variables change between zero and unity, it is risky not to use equation (5.7), or a combination of relative and absolute error.⁶²

6.3 Effect of adsorption and phase behavior

In this section the effect of adsorption and phase behavior on the performance of semi-discrete methods is examined. The

difference in input data between the runs presented in this section and the previous section is the adsorption and/or salinity. Adams' methods and Gear's method were no longer tested because of rather poor results obtained in the previous runs and the difficulty in changing the program to deal with the irreversibility of adsorption.

Table 6.5 shows the results for Type II(-) phase behavior environment with adsorption. Data set B in Table 6.1c was used. Compared with the previous case, IEVA and NREJ increased for both RK1(2) and RK1. Furthermore, the history of the time step size in Figure 6.23, which shows the results of Run 333, exhibit more frequent oscillation than previously.

Tables 6.6 and 6.7 present the results for Type II(+) phase behavior with no adsorption (data set C in Table 6.1c) and with adsorption (data set D in Table 6.1c), respectively. RK1 was less efficient for both cases compared with Type II(-) runs. Figures 6.24 and 6.25 show the performance of RK1 with $ERR = 0.01$ for each case.

Table 6.8 shows the comparison of FDE and RK1 for Type III phase behavior with and without adsorption (data set F and E). Figures 6.26 and 6.27 show the results of RK1 without adsorption and with adsorption, respectively. Hirasaki's relative permeability model was used for all these runs.

Table 6.9 summarizes the effect of adsorption and phase behavior on the efficiency of RK1. All runs were made with $ERR = 0.01$.

The computation time increased as the phase behavior environment was changed from Type II(-) to Type II(+) to Type III.

Since truncation error for RK1 is rather large compared with RK1(2), a question arises about the effect of the change in the time step size Δt_D on the numerical dispersion. The numerical dispersion for RK1 is expressed by exactly the same equation as for FDE because the time integrations are identical. When Δt_D is small compared with Δx_D , Δt_D can be neglected as in equation (3.7). For larger Δt_D , however, Δt_D may affect the numerical dispersion. Tables 6.3 and 6.5 through 6.8 suggest oil recovery is not very sensitive to the change in time step size when the phase behavior is Type II(-), whereas the time step did have a small effect on the oil recovery for the Type II(+) case. For the Type III case, the oil recovery was 100% for all runs.

6.4 Effect of each component on time step size selection

This section contains additional discussion about the time step size selection. All the results which have been presented so far were obtained by checking the error for all components. The effect of each component on step size control in chemical flooding is first discussed. Then simpler problems such as waterflooding and miscible displacement (single phase flow) are examined. Only RK1 was used and ERR was fixed to be 0.01. Thus Run 200 was considered as a reference run for all of the other chemical flooding runs. A summary of the results is presented in Table 6.10. Figures

6.28 to 6.34 show the total concentration history at the top and the time step size history at the bottom. Figures 6.28 to 6.32 show chemical flooding results. Figure 6.33 shows a waterflood. Figure 6.34 shows a miscible flood.

First, the injection of tracer was eliminated in Run 265 (Figure 6.28). Compared with Run 200 (Figure 6.20), a difference in the time step size and its amplitude of oscillation before surfactant breakthrough can be observed. Thereafter, the time step size is identical to Run 200.

In Run 267 (Figure 6.30), the error was checked only for the oil concentration (\tilde{C}_2). The history of the time step size is quite different from the previous two runs at two different times. One major difference is before 0.20 PV injection. Run 267 takes a time step size of around 0.002 PV from the very beginning, whereas in the previous two runs the time step size increased much more gradually. The other difference is after surfactant breakthrough. The time step size jumps to DTMAX in Run 267, while in the previous two runs it remained below DTMAX much longer.

The error was checked only for the surfactant concentration (\tilde{C}_3) in Run 268 (Figure 6.31). Before surfactant breakthrough, the time step size gradually increased with little oscillation. The time step size coincides with the upper edge of the oscillating time step size in Run 265. After surfactant breakthrough, the time step size was identical to Run 265.

In Run 269 (Figure 6.32), only the error in the polymer

concentration (\tilde{C}_4) was used to control the time step size. The time step size before surfactant breakthrough was almost the same as in Run 268 and increased to DTMAX thereafter. The oscillation in the concentration history was a little smaller than in Run 268. This may have been because the time step size before surfactant breakthrough was a little smaller in Run 269 than in Run 268. Run 269 and Run 267 suggest that the time step size should be equal to DTMAX (0.01) after surfactant breakthrough.

Since Run 267 which checked only the error of the oil component gave the best result, the same test was done for Type II(-) phase behavior with adsorption (Run 522) and Type III phase behavior with no adsorption (Run 446). A comparison is made with the runs which checked the errors of all the components in Table 6.11, and plotted results for Run 522 and 446 are presented in Figures 6.35 and 6.36, respectively. When the error was checked only for \tilde{C}_2 (oil), a larger time step size was taken and the quality of the result was still good.

Run 229 (Figure 6.33) is the result of waterflooding with no tracer injected. In this case the total concentration of water and oil is identical to water cut and oil cut. Initial condition was changed to residual water saturation and 100% water (no surfactant, no oil) was injected continuously. In this case the problem reduces to the well known waterflooding equation.

$$\frac{\partial S_w}{\partial t_D} = - \frac{\partial f_w}{\partial X_D} \quad (6.7)$$

Since the finite difference approximations used are the forward Euler in time and backward in space, the stability criterion is¹⁷

$$\frac{\Delta t_D}{\Delta X_D} \frac{df_w}{dS_w} \leq 1 \quad (6.8)$$

The maximum value of df_w/dS_w is obtained from the value at the flood front. Based on that maximum value, the stability requirement can be calculated as $\Delta t_D \leq 0.006$. Figure 6.33 shows Δt_D was about 0.001 before water breakthrough.

Run 260 (Figure 6.34) is a miscible displacement. The initial condition was 100% water with no tracer and 100% water with tracer was injected continuously. Since there exists only a single phase, df_w/dS_w in equation (6.8) is replaced by unity for the stability criterion. Thus, 40 grid blocks gives the stability requirement of $\Delta t_D \leq 0.025$. However, Figure 6.34 shows that a much smaller time step size was computed by RK1.

From Runs 229 and 260, it becomes clear that with RK1 the stability requirement was detected only when the time step size exceeded the stability limit. In other words, error tolerance was too small for these two runs. If larger error tolerance was used, RK1 should have detected the stability limit and selected a time step size around the stability limit.

Thus, the time step size history in Run 200 (Figure 6.20) is explained as follows. At the very beginning, a very small time step size was taken and it increased gradually. This behavior is due to the truncation error associated with tracer which had the steepest front. When the tracer was not injected (Run 265, Figure 6.28), larger time step was selected but still similar trend was observed. This is because of surfactant and polymer which also had steep (although less than tracer) front. When a steep front exists, a large truncation error is produced and RK1 selects a small time step. When the error was checked only for water (Run 266, Figure 6.29) or oil (Run 267, Figure 6.30) component, which did not have a steep front, such a small time step was not selected. From the fact that Run 266 and 267 did not show significant oscillation, the time step size at the beginning in Run 200 was smaller than necessary if only stability was desired. The reason the time step size increased gradually in Runs 200 and 265 is that the front of tracer, surfactant, and polymer got more and more smeared due to numerical dispersion.

After a while in Run 200, the time step size history began to oscillate with larger amplitude as the time step size reached the stability limit. When the time step size exceeded the stability limit, it was automatically reduced, however, it is increased again because of the rather small truncation error in the stable region.

After surfactant breakthrough, the time step size in Run 200

remained below DTMAX much longer than in Runs 266 and 267.

This is again the effect of surfactant. The truncation error at the tail of surfactant made RK1 select smaller time step size than DTMAX or stability limit.

The conclusions of this section are as follows. RK1 is controlling truncation error as it should be. The time step size is controlled by the component which has the steepest front. One important fact is that stability and truncation error are different problems. When an ODE integrator is used to control error, it also controls stability. If, however, one desires only stability, the ODE integrator may select a time step size smaller than necessary at times.

6.5 Additional test runs and summary of RK1(2) and RK1 for Type II(-) phase behavior environment without adsorption

Some additional test runs were made to attempt to reduce the computation time. So far the parameters PCT and YBIAS have been fixed to be 0.8 and 10^{-3} , respectively. YBIAS was increased to unity, which makes the error tolerance an absolute error. PCT was reduced to 0.25 or 0.50 because a larger PCT causes the rejection of the predicted time step too often, and also give a worse solution. Some results of both RK1(2) and RK1 are presented in Table 6.12. Plotted results are presented only for Runs 629 and 647 in Figures 6.37 and 6.38, respectively. With YBIAS = 1.0, the combination

of $PCT = 0.25$ and $ERR = 0.0001$ (Run 629) gave the best result using RK1(2) for both computer time and the quality of the solution. For RK1, $PCT = 0.25$ and $ERR = 0.01$ (Run 626) was the best.

A summary of RK1(2) and RK1 results for Type II(-) phase behavior without adsorption is given in Table 6.13. Only the best results are compared. If only oscillation rather than truncation error is used as the criterion of goodness, the error may be checked only for the oil component, or the absolute error may be checked with small PCT, and the computation cost will be less.

6.6 Stability requirement before surfactant breakthrough

In this section, the structure of the Jacobian matrix we are dealing with and its desirable features are first introduced. Then some eigenvalues before surfactant breakthrough are presented to derive the stability requirement. Although the analysis is limited to this special stage, it is important because this is the specific time period which imposes the most strict limitation on the time step size.

In order to analyze the stability, we have to express the system of equations we are dealing with in the same fashion as the test equation (5.18).

$$\dot{\underline{y}} = \underline{J}\underline{y} \quad (6.9)$$

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial y_1} & \frac{\partial F_1}{\partial y_2} & . & . & . & \frac{\partial F_1}{\partial y_N} \\ \frac{\partial F_2}{\partial y_1} & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{\partial F_N}{\partial y_1} & . & . & . & . & \frac{\partial F_N}{\partial y_N} \end{bmatrix} \quad (6.10)$$

where J is the Jacobian matrix and F is the derivative which is identical to the right hand side of equation (3.10). N is the total number of equations, which is the product of the number of components (NCOMP) and the number of grid blocks. \tilde{y} and \tilde{y}' are vectors of length N . Since we are using only the convection term, and it is approximated by backward difference, the structure of the Jacobian is as follows:

$$J = \begin{bmatrix} J_{11} & & & & & & & & & \\ J_{21} & J_{22} & & & & & & & & 0 \\ & J_{32} & J_{33} & & & & & & & \\ & & . & . & & & & & & \\ & & & . & . & & & & & \\ & & & & . & & & & & \\ & & & & J_{kk-1} & J_{kk} & & & & \\ & 0 & & & & . & . & . & . & \\ & & & & & & . & . & . & \\ & & & & & & & J_{II-1} & J_{II} & \end{bmatrix} \quad (6.11)$$

where J_{kk} and J_{kk-1} are all NCOMP by NCOMP block matrices. Thus, the Jacobian is a block lower triangular matrix.

For such a block triangular matrix with its diagonal blocks all being square matrices, it is possible to prove that the eigenvalues of the diagonal block matrix J_{kk} are also the eigenvalues of the matrix J .

Before surfactant breakthrough, J_{kk} ahead of the surfactant front (downstream) can be expressed

$$J_{kk} = \begin{bmatrix} \frac{\partial F_{m+1}}{\partial y_{m+1}} & \frac{\partial F_{m+1}}{\partial y_{m+2}} & 0 & 0 & 0 & 0 \\ \frac{\partial F_{m+2}}{\partial y_{m+1}} & \frac{\partial F_{m+2}}{\partial y_{m+1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial F_{m+5}}{\partial y_{m+1}} & \frac{\partial F_{m+5}}{\partial y_{m+2}} & 0 & 0 & \frac{\partial F_{m+5}}{\partial y_{m+5}} & 0 \\ \frac{\partial F_{m+6}}{\partial y_{m+1}} & \frac{\partial F_{m+6}}{\partial y_{m+2}} & 0 & 0 & 0 & \frac{\partial F_{m+6}}{\partial y_{m+6}} \end{bmatrix} \quad (6.12)$$

where

$$m = (k-1) * \text{NCOMP}$$

and the subscript $m+i$ indicates component i at k th block.

In the submatrix J_{kk} above, the alcohol (component 7) is not included, because it does not affect the eigenvalues. This is because it only adds zero elements in the seventh column and the seventh row. Thus submatrix J_{kk} is again a block lower triangular matrix and its eigenvalues are obtained from the following matrices:

$$A = \begin{bmatrix} \frac{\partial F_{m+1}}{\partial y_{m+1}} & \frac{\partial F_{m+1}}{\partial y_{m+2}} \\ \frac{\partial F_{m+2}}{\partial y_{m+1}} & \frac{\partial F_{m+2}}{\partial y_{m+2}} \end{bmatrix} \quad (6.13)$$

$$B = \begin{bmatrix} \frac{\partial F_{m+5}}{\partial y_{m+5}} & 0 \\ 0 & \frac{\partial F_{m+6}}{\partial y_{m+6}} \end{bmatrix} \quad (6.14)$$

Since there is only a water and an oil phase present (no microemulsion phase) ahead of the surfactant front, then

$$F_{m+1} = -\frac{1}{\Delta x_D} \{ (f_1 C_{11})_k - (f_1 C_{11})_{k-1} \} \quad (6.15a)$$

$$F_{m+2} = -\frac{1}{\Delta x_D} \{ (f_2 C_{22})_k - (f_2 C_{22})_{k-1} \} \quad (6.15b)$$

$$F_{m+5} = -\frac{1}{\Delta x_D} \{ (f_1 C_{51})_k - (f_1 C_{51})_{k-1} \} \quad (6.15c)$$

$$F_{m+6} = - \frac{1}{\Delta x_D} \{ (f_1 c_{61})_k - (f_1 c_{61})_{k-1} \} \quad (6.15d)$$

where

$$c_{11} = c_{22} = 1.0 \quad (6.16)$$

$$c_{51} = c_5 / c_1 \quad (6.17)$$

$$c_{61} = c_6 / c_1 \quad (6.18)$$

$$c_1 = S_w \quad (6.19)$$

$$c_2 = 1 - S_w \quad (6.20)$$

the matrices A and B can be rewritten

$$A = - \frac{1}{\Delta x_D} \frac{df_w}{dS_w} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (6.21)$$

$$B = - \frac{1}{\Delta x_D} \frac{df_w}{dS_w} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.22)$$

Their eigenvalues are

$$\lambda = -\frac{2}{\Delta x_D} \frac{df_w}{dS_w}, \quad -\frac{1}{\Delta x_D} \frac{f_w}{S_w}, \quad \text{or } 0 \quad (6.23)$$

and these are also eigenvalues of the Jacobian matrix (6.10).

If we consider the RK1 method, the stability requirement for the blocks ahead of the surfactant front is obtained from eigenvalues (6.23) and the stability region shown in Figure 6.39. Taking the eigenvalue largest in magnitude

$$\frac{2\Delta t_D}{\Delta x_D} \frac{df_w}{dS_w} \leq 2 \quad (6.24)$$

Thus the stability requirement is

$$\Delta t_D \leq \frac{\Delta x_D}{\frac{df_w}{dS_w}} \quad (6.25)$$

This stability criterion agrees with the von Neumann stability analysis for the waterflood equation when fully discretized using backward difference approximation in space and forward difference in time.

Considering the case of Type II(-) phase behavior with no adsorption, the water saturation at the oil bank from the simulated result is about 0.57. The derivative df_w/dS_w at this saturation is about 7.8, which gives the stability requirement of $\Delta t_D \cong 0.003$.

The stability requirement estimated from the RK1 and FDE runs before surfactant breakthrough is also $\Delta t_D \approx 0.003$. The RK1 selected the time step size of around 0.003 (Figure 6.30). The FDE gave oscillating solution when $\Delta t_D \geq 0.003$ (Figure 6.13). From this agreement it is conjectured that the stability before surfactant breakthrough is governed by the slope of fractional flow curve at the oil bank saturation.

When the dispersion term is introduced like in equation (3.1), the system of equations can be written

$$\tilde{y}' = J_T \tilde{y} \quad (6.26)$$

where the Jacobian matrix involves the effect of both convection and dispersion

$$J_T = J_C + J_D \quad (6.27)$$

The Jacobian matrix J_C obtained from convection term is block lower triangular as before, whereas J_D from dispersion is block tridiagonal. Thus, the summed Jacobian matrix J_T is block tridiagonal and it may seem impossible to use the advantage of the block triangular matrix. However, the Jacobian matrix J_T still can be divided into a block lower triangular matrix in a somewhat different way by taking into account the fact that $\partial C_{ij} / \partial x_D$ is zero beyond some point (i.e. in the oil bank).

$$J_T = \begin{bmatrix} J_{11} & J_{12} & & & & & & & & & \\ J_{21} & J_{22} & J_{23} & & & 0 & & & & & \\ & J_{32} & J_{33} & J_{34} & & & & & & & \\ & & \cdot & \cdot & \cdot & & & & 0 & & \\ & & & \cdot & \cdot & \cdot & & & & & \\ & 0 & & & J_{i,i-1} & J_{i,i} & J_{i,i+1} & & & & \\ & & & & & J_{i+1,i} & J_{i+1,i+1} & & & & \\ & & & & & & J_{i+2,i+1} & J_{i+2,i+2} & & 0 & \\ & & & & & & & J_{i+3,i+2} & J_{i+3,i+3} & & \\ & & & & & & & \cdot & \cdot & & \\ & & 0 & & & & & & \cdot & \cdot & \\ & & & & & & 0 & & & \cdot & \\ & & & & & & & & & J_{I,I-1} & J_{I,I} \end{bmatrix}$$

(6.28)

Then, the eigenvalues of the submatrix $J_{k,k}$ ($k = i+2, \dots, I$) in the oil bank are still eigenvalues of the whole matrix J_T at the same time. However, we can no longer conjecture the eigenvalues still dominate the ones upstream, since the effect of dispersion there is proportional to $\alpha_{Dj}/(\Delta X_D)^2$, where α_{Dj} is the dispersivity coefficient in equation (3.1). For some sufficiently large α_{Dj} or small ΔX_D , the dispersion term behind surfactant front may dominate the stability requirement.

6.7 Analysis of stability at blocks where surfactant is present

Although stability was analyzed in the previous section for the oil bank where there is no surfactant, the stability at blocks where there is surfactant was not analyzed because of the difficulty of obtaining the Jacobian matrix and its eigenvalues.

In this section, a stability criterion for the waterflood equation is introduced and the stability is analyzed for such blocks. An attempt is made to explain why oscillation occurs in the production history just before and after surfactant breakthrough (Figure 6.9), even when considerably smaller time steps are employed compared to the previous analysis.

Because the nature of equation (3.5) is similar to the waterflood equation, the stability of the waterflood equation below is examined.

$$\frac{\partial S_1}{\partial t_D} + \frac{\partial f_1}{\partial x_D} = 0 \quad (6.29)$$

This equation can be derived from equation (3.5) if we assume that the change in C_{ij} with respect to both time and space is negligible or that we have a sharp surfactant front and C_3 changes from the initial concentration to the injected concentration.

When equation (6.29) is approximated by backward difference in space and forward difference in time as is done in the simulator, a von Neumann stability analysis gives the stability criterion as follows:

$$f' \frac{\Delta t_D}{\Delta x_D} (1 - \cos\theta) (f' \frac{\Delta t_D}{\Delta x_D} - 1) \leq 0 \quad (6.30)$$

$$0 \leq \theta \leq 2\pi$$

where

$$f' = \frac{df}{dS} \quad (6.31)$$

Although f' is not constant in an actual problem, it was assumed to be constant in the derivation of equation (6.30). Then, the approximation to f' is

$$f' = \frac{f_k - f_{k-1}}{S_k - S_{k-1}} \quad (6.32)$$

where the subscript k designates block number. From equation (6.30), the stability criterion is given by

- 1) $f' \Delta t_D \leq \Delta x_D$ if f' is positive
- 2) always unstable if f' is negative

Although the criterion 2) never arises in waterflooding because f' is always positive, f' can be negative in micellar flooding due to the change in the residual saturation (Figure 6.5) as an effect of surfactant. Table 6.14 shows the aqueous phase profile of Run 207 (fully discrete, $\Delta t = 0.001$, Type II(-), no adsorption) at 0.5 PV injection. f' is presented in the bottom line and a negative value appears at dimensionless distance of 0.9 from injector just upstream of the surfactant front. The appearance of a negative f' can be explained by comparing two fractional flow curves obtained from each of the two blocks. Figure 6.40 shows an example. Since the residual saturation is different for the two blocks, the relative permeability curves are also different, which yields two different fractional flow curves, one for each block. It was also confirmed that a negative value of f' appeared intermittently somewhere behind the surfactant front. The order of the negative value could be as large as 10^2 .

Although the stability requirement (6.30) was derived for the finite different approximation backward in space and forward in time, the same can be true for the formulation of RK1(2).

$$y_{i+1} = y_i + \Delta t \left(\frac{1}{256} f_0 + \frac{255}{256} f_1 \right) \quad (5.9)$$

Recall that the subscripts i and $i+1$ here designate the time level. Although the time integration is different from the forward Euler, the derivative f_1 is obtained using forward Euler

$$f_0 = f(t_i, y_i) \quad (5.11)$$

$$f_1 = f(t_i + \frac{1}{2}\Delta t, y_i + \frac{1}{2}\Delta t f_0) \quad (5.12)$$

Thus, a larger error is introduced to f_1 and consequently y_{i+1} in equation (5.9), which is affected by the error if the stability condition (6.30) is not satisfied.

From Figure 6.40 it can be deduced that if the difference between the two fractional flow curves is small compared with the difference between the saturations of each block, the negative f' can be eliminated. One suggestion can be made at this point. Since fractional flow depends on relative permeabilities, which are functions of the residual saturations, we should look over the way we determine residual saturations. In the simulator, residual saturations are given by equations (3.38). These equations give a linear relationship between the residual saturations and the logarithm of the capillary number. Compared with the experimental data, such as the ones shown in Figure 3.7, this functional relationship may yield too much change in the residual saturation at surfactant front, which causes the large change in the fractional flow. From the viewpoint of both numerical stability and experimental data fitting, it is suggested that equation (3.38) for the non-wetting phase be changed to another form.⁶ However,

it is impossible to eliminate this instability completely as long as the continuity equations are solved explicitly, because the change in the fractional flow curve is an essential part of the micellar/polymer flooding.

One question, however, arises. Why is the fractional flow profile so smooth as shown in Figure 6.3 in spite of the unstable condition? This is because the unstable condition occurs only locally and temporarily, although continually. Even if the unstable condition arises, it becomes stable in a short time and the large error produced in the unstable condition may later die out. Furthermore, the error that propagates downstream may also die out due to the stable condition existing there. If, however, the unstable condition occurs near production block, the error can reach the producer before it dies out and causes an oscillation in production history.

When the semi-discrete method is used, the time step size is usually increased to DTMAX after surfactant breakthrough. Sometimes, however, it is decreased again for a short time, then goes up to DTMAX and becomes stable (Figure 5.30). This drop in time step size is also attributed to the instability which has been mentioned above.

Although less significant compared with the effect of surfactant, the polymer also changes the shape of the fractional flow curve by changing the viscosity. Therefore, it may also be necessary to evaluate the effect of polymer when one considers numerical stability.

Table 6.1a Basic input data used to test semi-discrete method.**

VT = 2.0	FFDV = 0.04	NCOMP = 6	ICT = 40	ICTL = 1	ICTU = 40
UT = 0.0	ABPERM = 1.0	PHI = 0.2	EPHI3 = 1.0	EPHI4 = 1.0	DISP = 0.0
C51I = *	C61I = 0.0	S1 = 0.63	S2 = 0.37	S1RW = 0.37	S2RW = 0.37
G11 = 6.785	G12 = -7.058	G13 = 0.11	G21 = 6.285	G22 = -7.058	G23 = 0.11
T11 = 0.37	T12 = 2.87	T21 = 0.37	T22 = 0.9	XIFTW = 1.3	
ALPHA1 = 0.0	ALPHA2 = 0.0	ALPHA3 = 50.0	ALPHA4 = 0.0	ALPHA5 = 0.0	
VIS1 = 1.0	VIS2 = 5.0	AP1 = 100.0	AP2 = 1000.0	AP3 = 10000.0	SSLOPE = 0.0
GAMHF = 13.6	POWN = 1.0	CSE1 = 0.00	RKMAX = 0.0	BRK = 0.0	
P1RW = 0.05	P2RW = 1.0	P3R = 1.0	E1 = 1.5	E2 = 1.5	E3 = 1.0
P1RC = 1.0	P2RC = 1.0	C3MAX0 = 0.3	C3MAX1 = 0.1	C3MAX2 = 0.3	
C2PLC = 0.0	C2PRC = 1.0	CSEL = 0.8	CSEU = 1.2		
QV = 0.0	RCSE = 1.0	AD31 = *	AD32 = *	B3D = 100.0	
A4D = *	B4D = 100.0	XK96 = 0.0	XK86 = 0.0	XKC = 0.0	XKHAT = 0.0

*Varied according to phase behavior or adsorption (see Table 6.1c).

**Explanation of input data is given in Appendix C.

Table 6.1b Basic input data used to test semi-discrete method.

Composition of injected slug

	Surfactant slug	Polymer buffer
Slug size (PV)	0.1	1.9
Water (vol. frac.)	0.9	1.0
Oil (vol. frac.)	0.0	0.0
Surfactant** (vol. frac.)	0.1	0.0
Polymer (wt. %)	0.1	0.1
Anion (normalized)	*	*
Tracer	1.0	1.0
Alcohol	0.0	0.0

*Varied according to phase behavior (see Table 6.1c).

**Combined with alcohol as an approximation.

Table 6.1c Basic input data used to test semi-discrete method.

Data set		A	B	C	D	E	F
Phase behavior		II(-)	II(-)	II(+)	II(+)	III	III
Adsorption		No	Yes	No	Yes	No	Yes
C51I	Initial anion concentration in water phase (normalized)	0.8	0.8	1.3	1.3	1.0	1.0
C5(1)	Anion concentration in surfactant slug (normalized)	0.7	0.7	1.3	1.3	1.0	1.0
C5(2)	Anion concentration in polymer buffer (normalized)	0.5	0.5	1.3	1.3	1.0	1.0
AD31	Adsorption parameter for surfactant	0.0	0.7	0.0	0.35	0.0	0.7
AD32	Adsorption parameter for surfactant	0.0	0.0	0.0	0.35	0.0	0.0
A4D	Adsorption parameter for polymer	0.0	0.7	0.0	0.7	0.0	0.7

Table 6.2 Comparison of FDE method and semi-discrete methods
with various ODE integrators (Type II(1), no adsorption).

Run No.	Fig. No.	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
203	6.11	FDE	($\Delta t_D = 0.002$)	1000	-	31.0	good	60.1
204	6.12	"	($\Delta t_D = 0.003$)	667	-	22.8	fair	60.2
206	6.13	"	($\Delta t_D = 0.004$)	500	-	18.5	poor	60.3
196	6.14	RK1(2)	0.00005	968	13	33.7	good	60.0
197	6.15	"	0.0001	734	4	27.0	fair	60.0
198	6.16	"	0.0002	740	44	27.6	poor	60.0
498	6.17	Adams'	0.0005	1217	-	43.1	good	60.1
496	6.18	"	0.001	1140	-	41.2	good	60.1
495	6.19	"	0.01	1051	-	38.9	fair	60.3

Table 6.3 Comparison of FDE and RK1 (Type II(-), no adsorption).

Run No.	Fig. No.	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
203	6.11	FDE	$\Delta t_D = 0.002$	1000	-	31.0	good	60.1
204	6.12	"	$\Delta t_D = 0.003$	667	-	22.8	fair	60.2
200	6.20	RK1	0.01	596	29	24.0	good	60.1
201	6.21	"	0.02	428	2	18.9	fair	60.1
202	6.22	"	0.05	356	7	16.7	poor	60.2

Table 6.4 Results of predictor-corrector methods at 0.1 PV injected.

Run No.	534	535	531
Method	Adams'	Adams'	Gear's
ERR	0.001	0.0001	0.0001
Max. Δt_D	7.4×10^{-3}	4.3×10^{-3}	3.2×10^{-3}
Min. Δt_D	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-6}
Average Δt_D	4.0×10^{-3}	1.6×10^{-3}	8.9×10^{-4}
Highest order q	2	2	3
IEVA	650	1792	3809
NSTEP	25	62	113
NJE	5	14	30
Net IEVA	50	112	209

Table 6.5 Comparison of FDE, RK1(2), and RK1 for Type II(-), with adsorption.

Run No.	Fig. No.	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
282	-	FDE	$\Delta t_D = 0.002$	1000	-	31.8	good	60.1
283	-	"	$\Delta t_D = 0.003$	667	-	23.3	fair	60.2
285	-	RK1(2)	0.00005	1116	5	38.4	good	60.0
287	-	"	0.0001	848	3	30.8	good-fair	60.0
333	6.23	RK1	0.01	725	52	27.6	good	60.1
332	-	"	0.02	549	36	22.3	fair	60.1

Table 6.6 Comparison of FDE and RK1 for Type II(+), no adsorption.

Run No.	Fig. No.	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
407	-	FDE	$\Delta t_D = 0.001$	2000	-	56.1	good	74.3
408	-	"	$\Delta t_D = 0.002$	1000	-	31.1	good	74.9
409	-	"	$\Delta t_D = 0.003$	667	-	22.6	fair	75.6
411	6.24	RK1	0.01	735	95	27.5	good	74.8
412	-	"	0.02	595	86	23.4	good-fair	75.1
413	-	"	0.05	476	65	19.8	poor	75.5

Table 6.7 Comparison of FDE, RK1(2), and RK1 for Type II(+), with adsorption.

Run No.	Fig. No.	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
406	-	FDE	$\Delta t_D = 0.001$	2000	-	56.1	good	51.3
384	-	"	$\Delta t_D = 0.002$	1000	-	31.0	good	51.8
385	-	"	$\Delta t_D = 0.003$	667	-	22.8	poor	53.4
395	6.25	RK1	0.01	1005	149	35.8	good	51.4
389	-	"	0.02	778	105	29.1	fair	51.7
391	-	RK1(2)	0.00005	1680	120	53.6	good	51.0

Table 6.8 Comparison of FDE and RK1 for Type III.

Run No.	Fig. No.	Adsorption	Method	ERR	IEVA	NREJ	CPU (sec.)	Quality	E_R (%)
445	-	No	Euler	$\Delta t_D = 0.001$	2000	-	55.5	good	100
444	-	"	"	$\Delta t_D = 0.002$	1000	-	30.9	fair	100
442	6.26	"	RK1	0.01	2760	766	86.9	good	100
464	-	Yes	Euler	$\Delta t_D = 0.001$	2000	-	58.7	good	100
465	-	"	"	$\Delta t_D = 0.002$	1000	-	32.6	fair	100
463	6.27	"	RK1	0.01	2594	466	84.0	good	100

Table 6.9 Effect of adsorption and phase behavior environment
on the efficiency of RK1 (ERR = 0.01).

Run No.	Fig. No.	Phase behavior	Adsorption	IEVA	NREJ	CPU (sec.)	Quality
200	6.20	II(-)	No	596	29	24.0	good
333	6.23	II(-)	Yes	725	52	27.6	good
411	6.24	II(+)	No	735	95	27.5	good
395	6.25	II(+)	Yes	1005	149	35.8	good
442	6.26	III	No	2760	766	86.9	good
463	6.27	III	Yes	2594	466	84.0	good

Table 6.10 Effect of each component on time step size selection
(RK1, ERR = 0.01, Type II(-), no adsorption).

Run No.	Fig. No.	Component checked	IEVA	NREJ	CPU (sec.)	Quality	E_R (%)
265	6.28	C_1-C_5	528	37	21.9	good-fair	60.1
266	6.29	C_1	423	51	18.3	good-fair	60.2
267	6.30	C_2	455	58	19.2	good	60.2
268	6.31	C_3	450	1	18.9	poor	60.1
269	6.32	C_4	419	1	18.0	poor	60.1
200	6.20	All	596	29	24.0	good	60.1

Table 6.11 Improvement in time step size selection by checking errors only for oil component (RK1, ERR = 0.01).

Run No.	Fig. No.	Phase behavior	Adsorption	Component checked	IEVA	NREJ	CPU (sec.)	Quality	E _R (%)
200	6.20	II(-)	No	A11	596	29	24.0	good	60.1
267	6.30	II(-)	No	C ₂	455	58	19.2	good	60.2
333	6.23	II(-)	Yes	A11	725	52	27.6	good	60.1
522	6.35	II(-)	Yes	C ₂	435	27	18.3	fair	60.2
442	6.26	III	No	A11	2760	766	86.9	good	100.0
446	6.36	III	No	C ₂	1933	528	60.8	good	100.0

Table 6.12 Test runs with YBIAS = 1.0 (RK1(2) and RK1, Type II(-), no adsorption).

Run No.	Fig. No.	Method	PCT	ERR	IEVA	NREJ	CPU (sec.)	Quality
626	-	RK1	0.25	0.01	522	0	21.2	good
627	-	"	"	0.1	336	0	15.5	fair
628	-	RK1(2)	"	0.00005	882	0	30.8	good
629	6.37	"	"	0.0001	742	0	26.8	good
630	-	"	"	0.001	572	0	22.0	poor
645	-	RK1	0.50	0.01	377	2	16.8	good-fair
646	-	RK1(2)	"	0.00005	654	9	24.6	good-fair
647	6.38	"	"	0.0001	616	3	23.5	poor

Table 6.13 Summary of RK1(2) and RK1 (Type II(-), no adsorption).

Run No.	Fig. No.	Method	Component checked	ERR	YBIAS	PCT	IEVA	NREJ	CPU (sec.)
203	6.11	FDE	-	$\Delta t_D = 0.002$	-	-	1000	-	31.0
196	6.14	RK1(2)	A11	0.00005	0.001	0.8	968	13	33.7
648	-	"	C ₂	"	"	"	814	72	29.0
629	6.37	"	A11	0.0001	1.0	0.25	742	0	26.8
200	6.20	RK1	A11	0.01	0.001	0.8	596	29	24.0
267	6.30	"	C ₂	0.01	0.001	0.8	455	58	19.2
626	-	"	A11	0.01	1.0	0.25	522	0	21.2





Table 6.14 Aqueous phase profile of Run 207 at 0.5 PV injection
(Type II(-), no adsorption).

X _D	0.850	0.875	0.900	0.925	0.950	0.975	
Saturation	0.5695	0.5610	0.5557	0.5550	0.5751	0.5760	0.5770
Fractional flow	0.6426	0.6320	0.6258	0.6315	0.6382	0.6479	0.6571
C ₁₃	0.9982	0.9990	0.9994	0.9997	0.9998	0.9999	1.000
C ₂₃	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C ₃₃	0.0018	0.0010	0.0006	0.0003	0.0002	0.0001	0.0
f'	1.247	1.170	-8.143	0.333	10.78	9.20	

X_D = Fractional distance

$$f' = \frac{f_k - f_{k-1}}{S_k - S_{k-1}}$$

Table 6.15 Legend for the total concentration history plots.

Symbol	Component
	Water
	Oil
+	Surfactant*
x	Polymer
	Anion
	Tracer

*For all total concentration history plots, surfactant concentration is five times amplified.

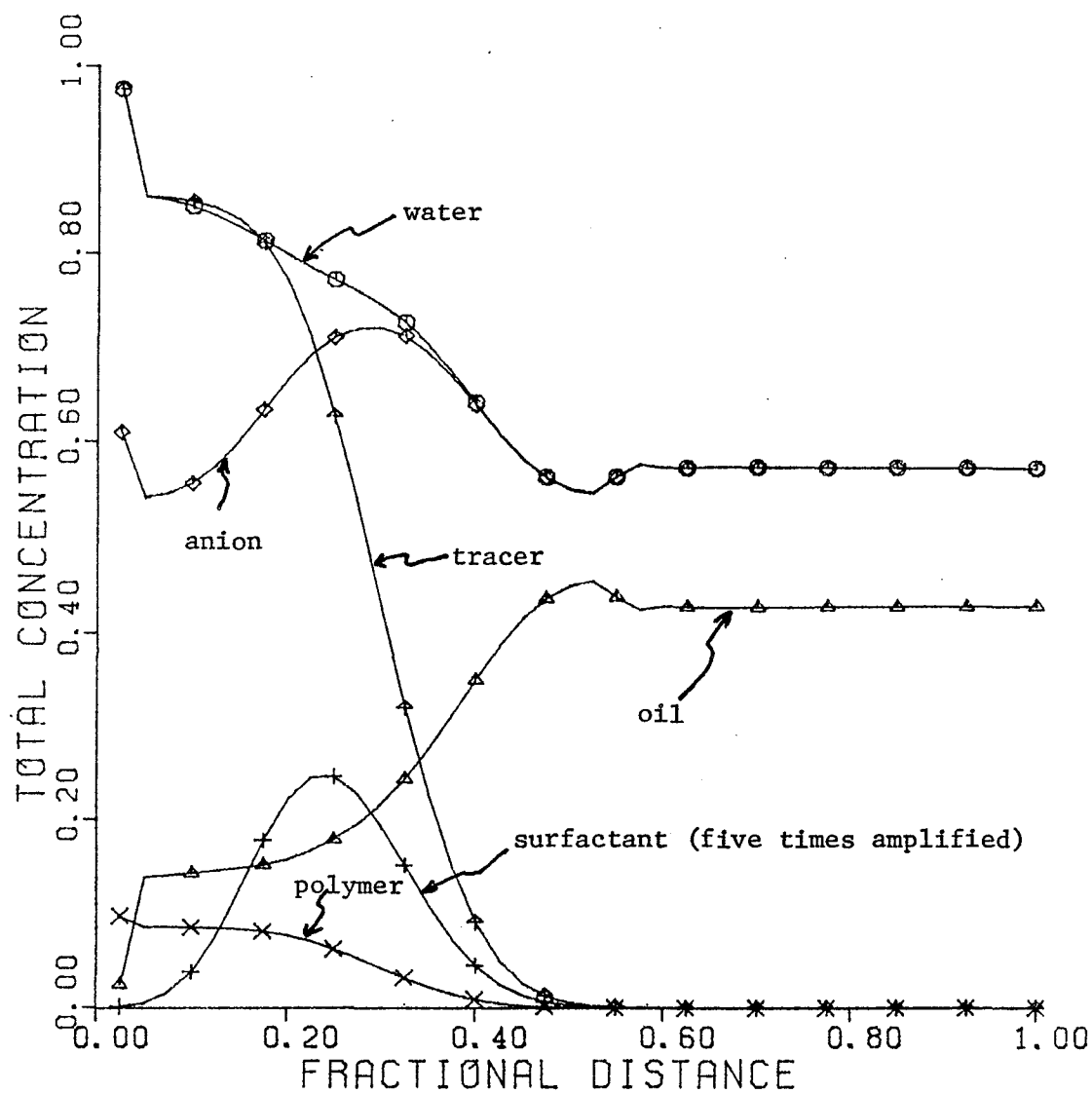


Figure 6.1. Total concentration profile at 0.25 PV for Run 207 (FDE method, $\Delta t_D = 0.001$ PV, Type II(-), no adsorption).

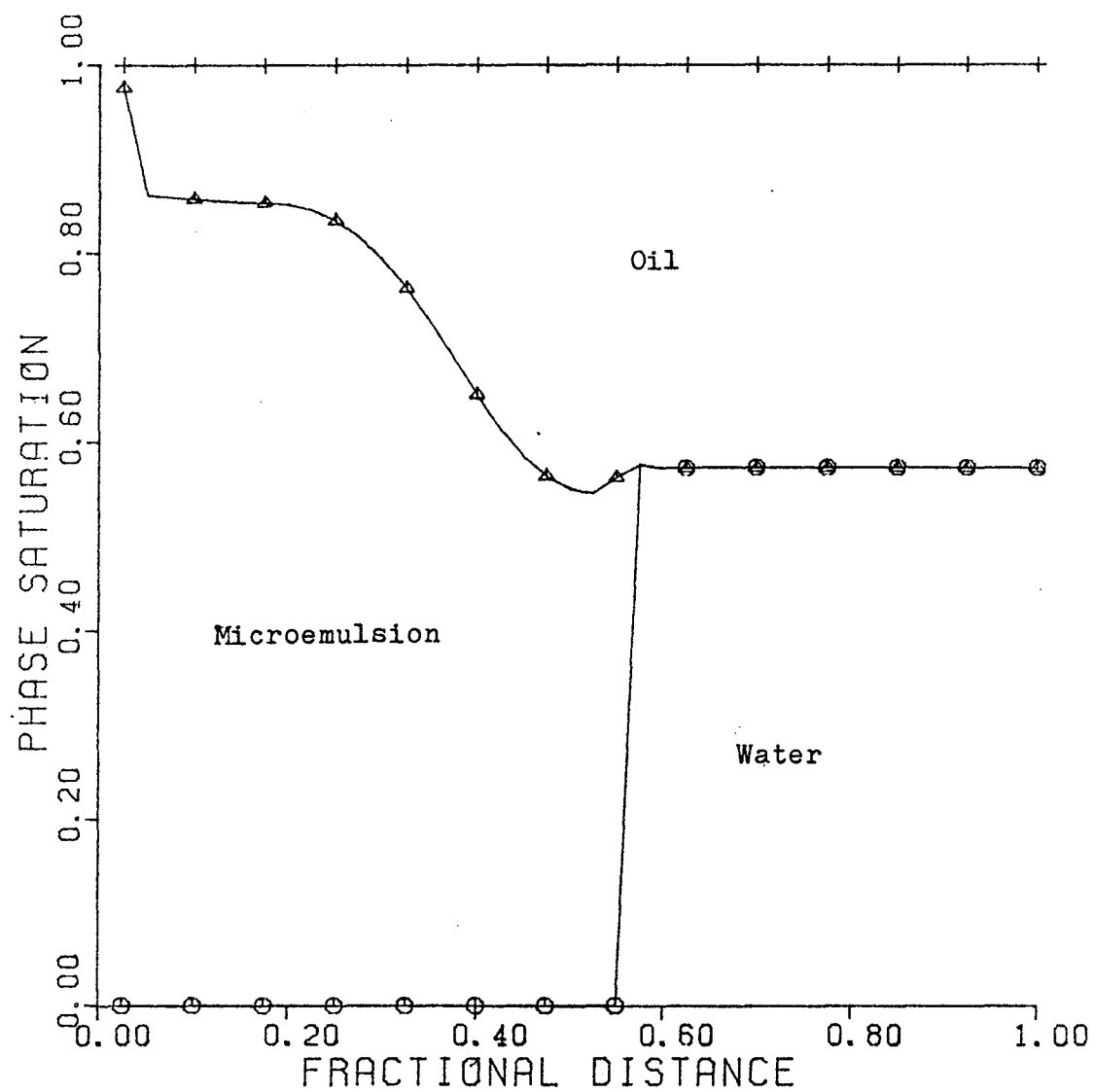


Figure 6.2. Phase saturation profile at 0.25 PV for Run 207.

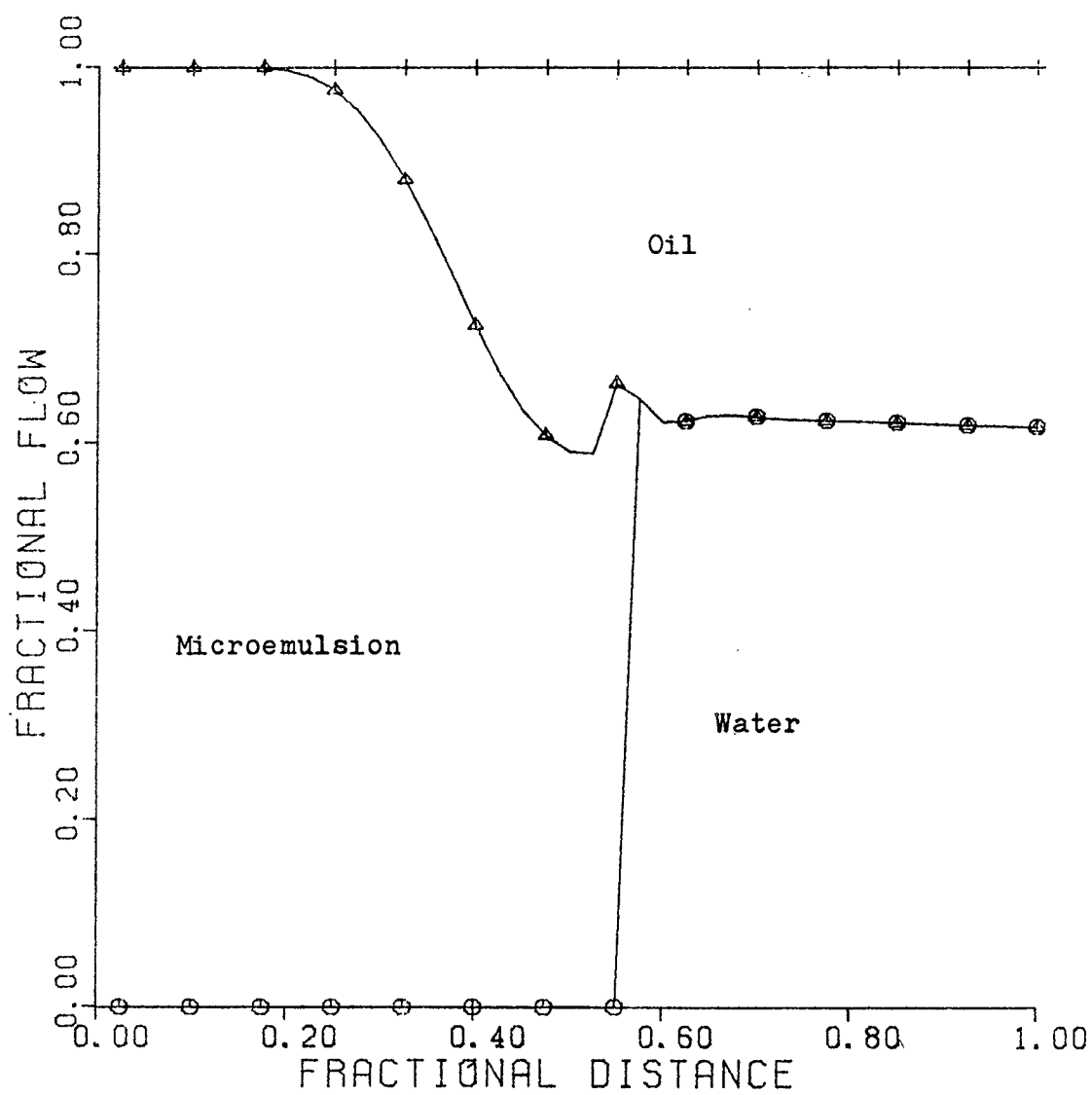


Figure 6.3. Fractional flow profile at 0.25 PV for Run 207.

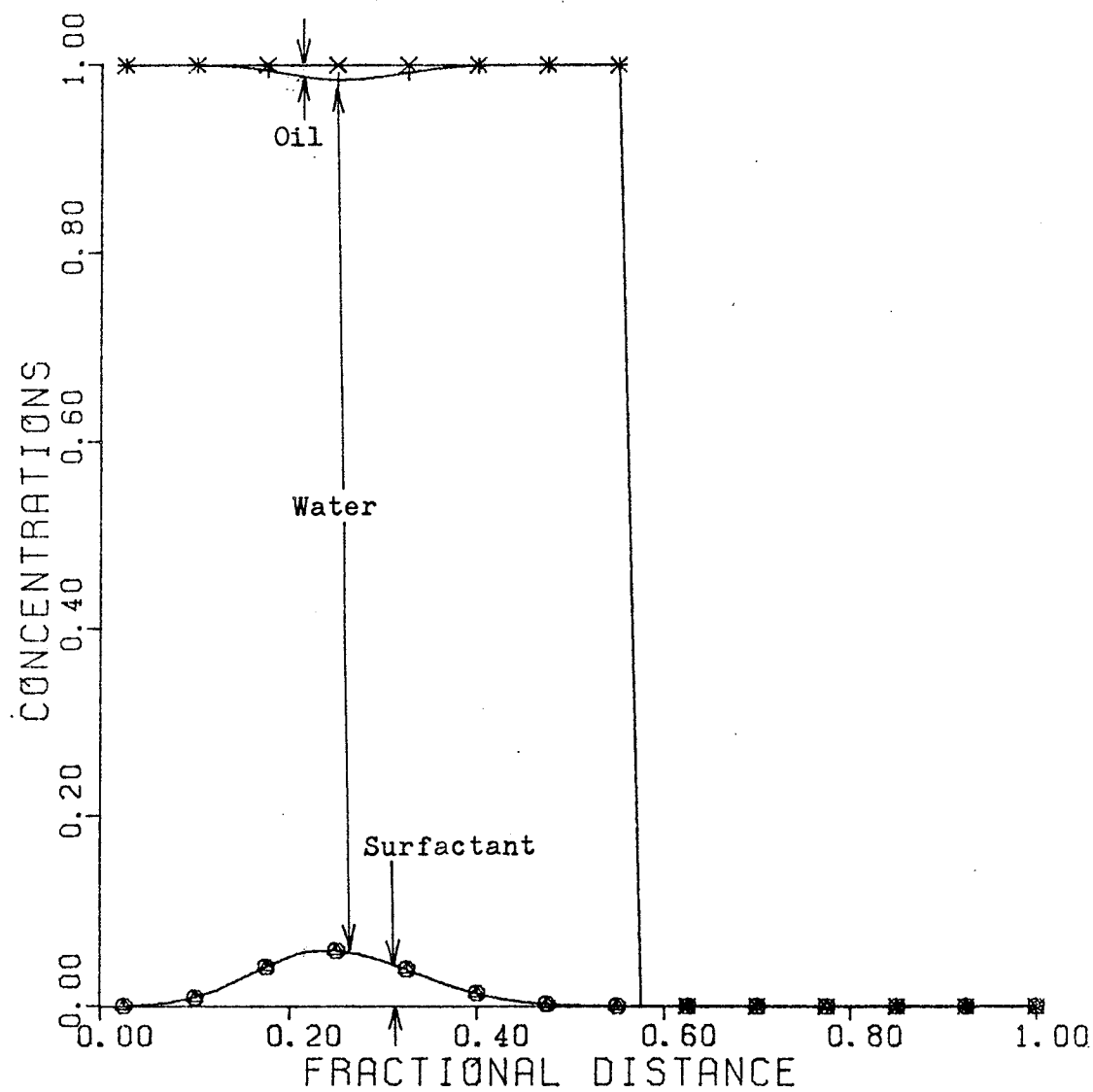


Figure 6.4. Microemulsion phase concentration profile at 0.25 PV for Run 207.

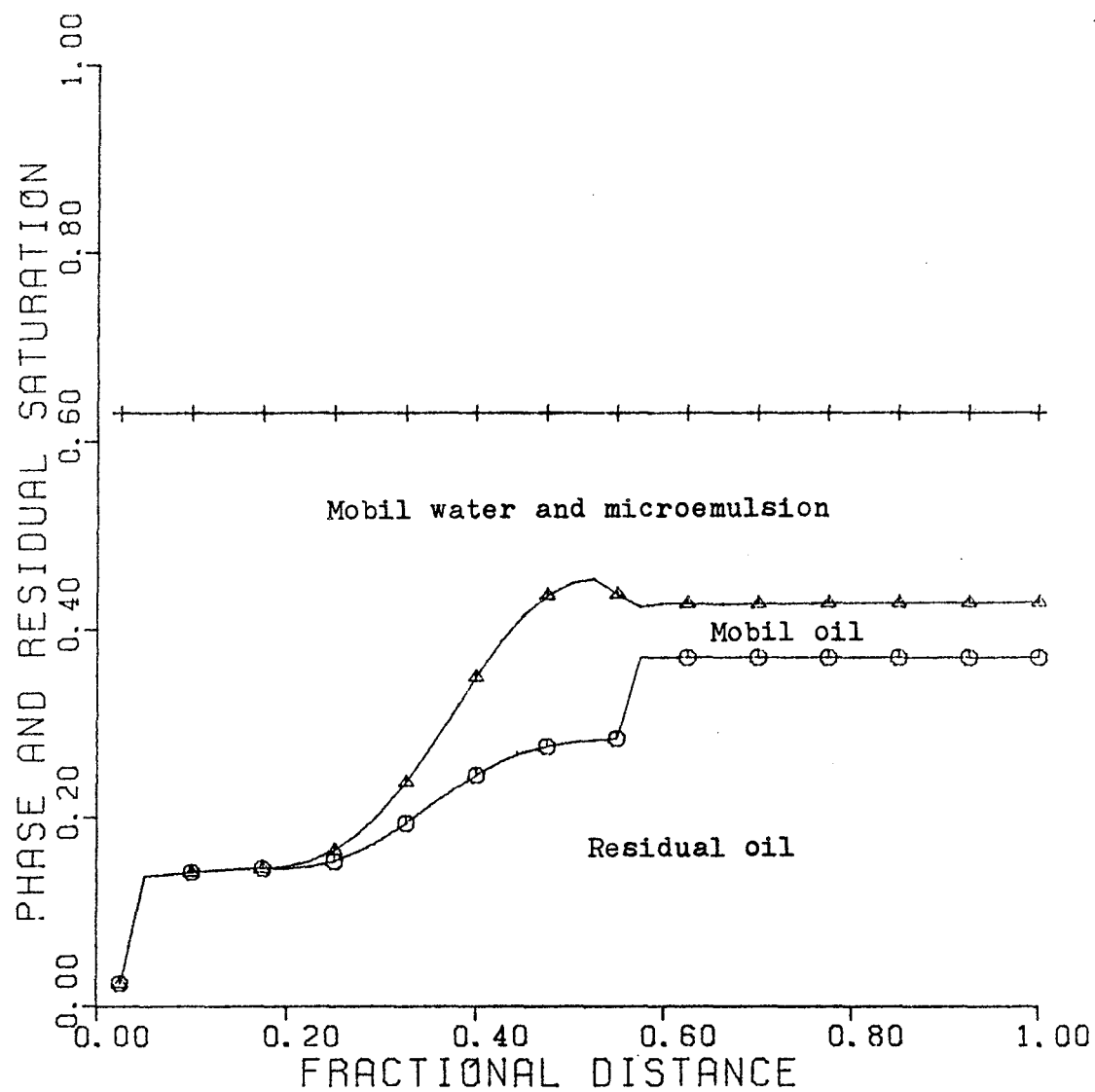


Figure 6.5. Oleic phase saturation and its residual saturation profile at 0.25 PV for Run 207.

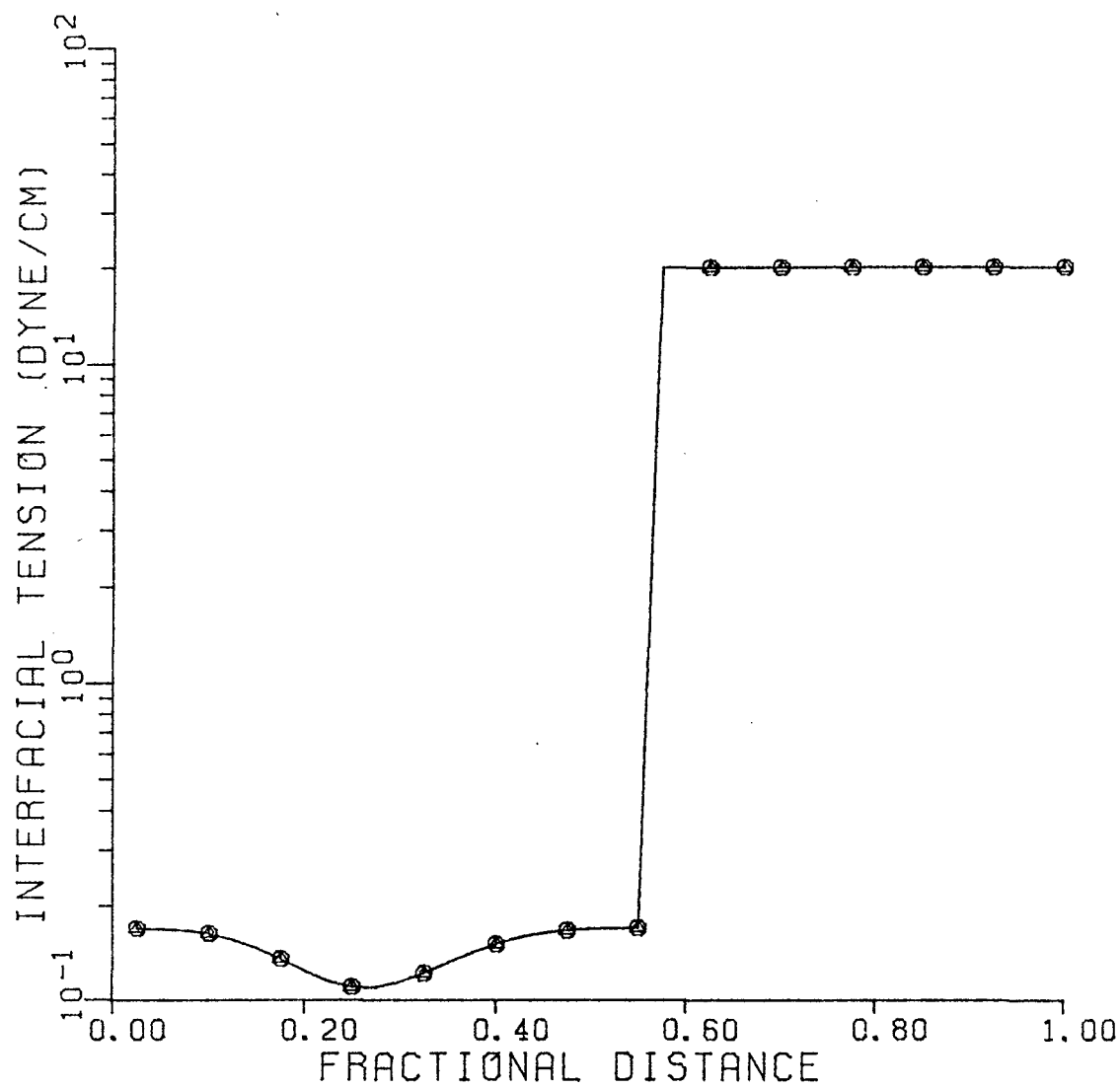


Figure 6.6. Interfacial tension profile at 0.25 PV for Run 207.

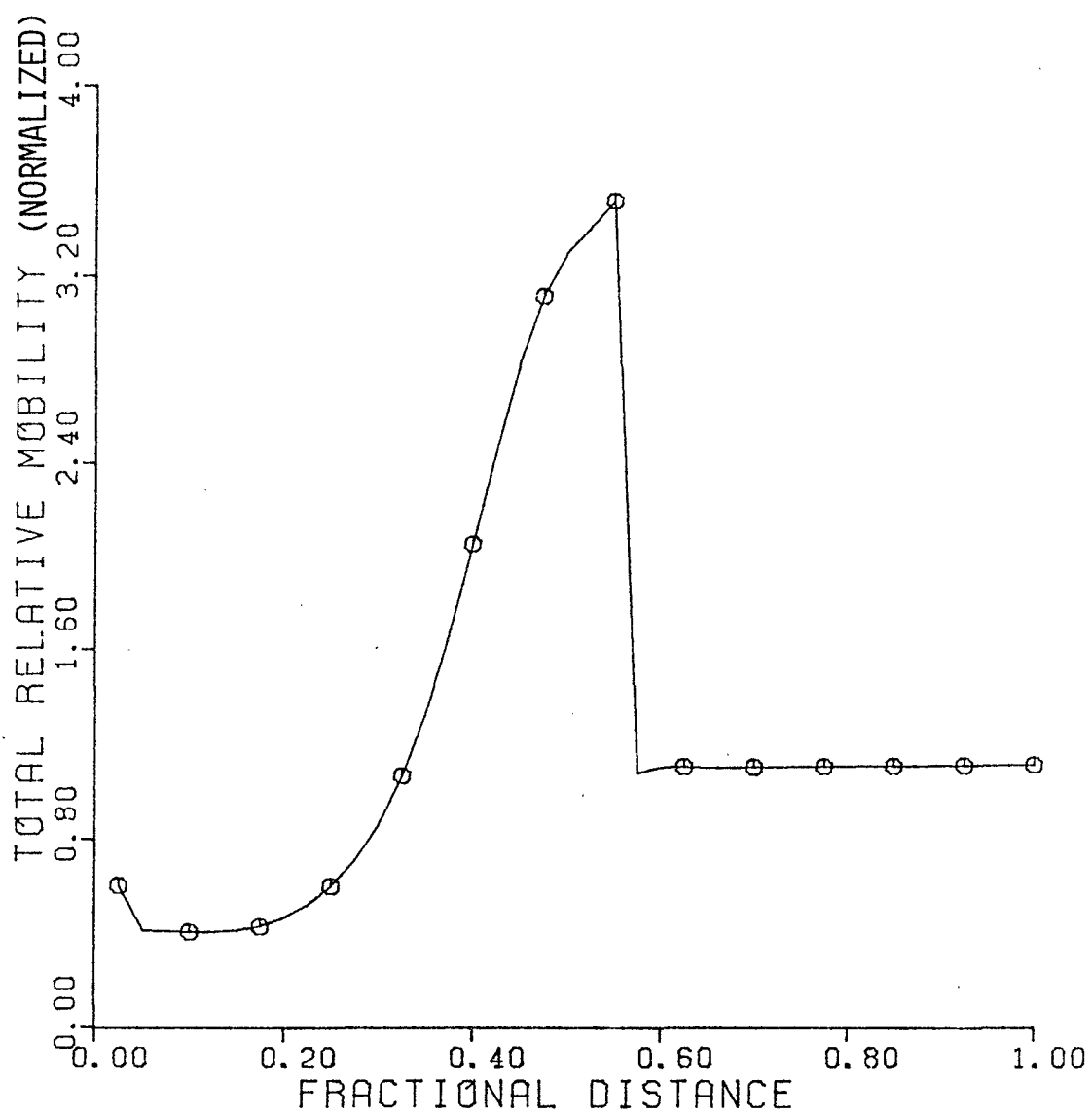


Figure 6.7. Total relative mobility profile at 0.25 PV for Run 207;

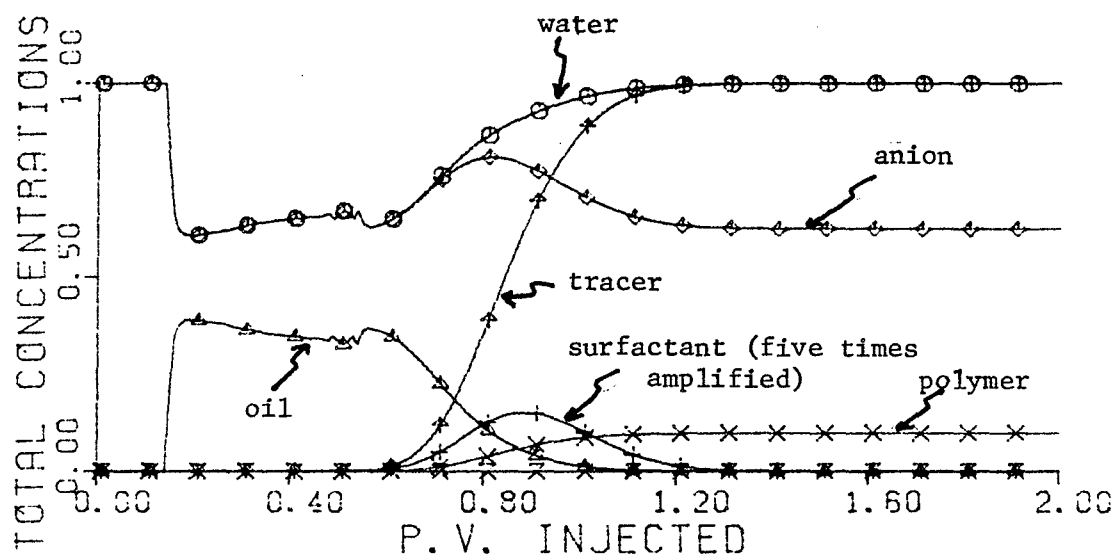


Figure 6.8. History of total concentration in effluent for Run 207.

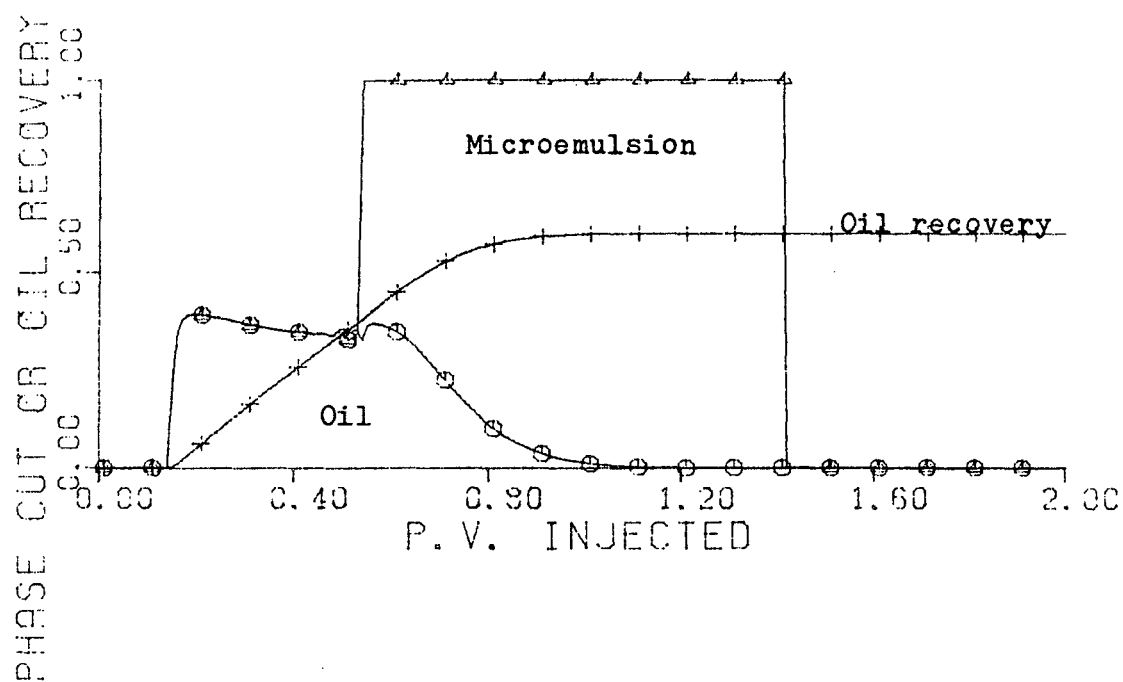


Figure 6.9. History of phase cuts and cumulative oil production for Run 207.

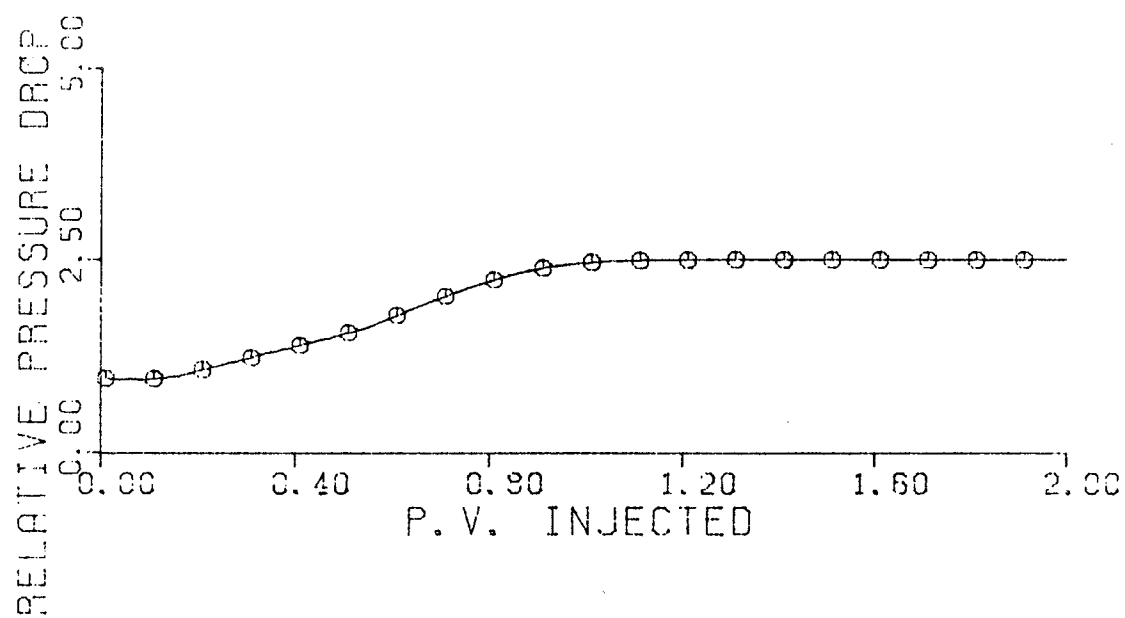


Figure 6.10. History of pressure drop (normalized) between producer and injector for Run 207.

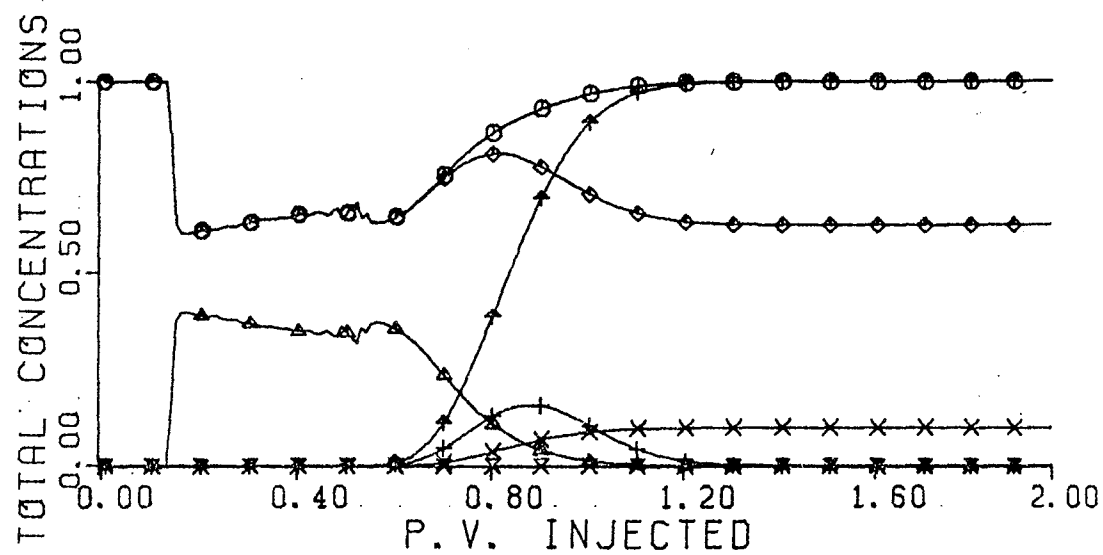


Figure 6.11. History of total concentration in effluent for Run 203

(FDE method, $\Delta t_D = 0.002$, Type II(-), no adsorption).

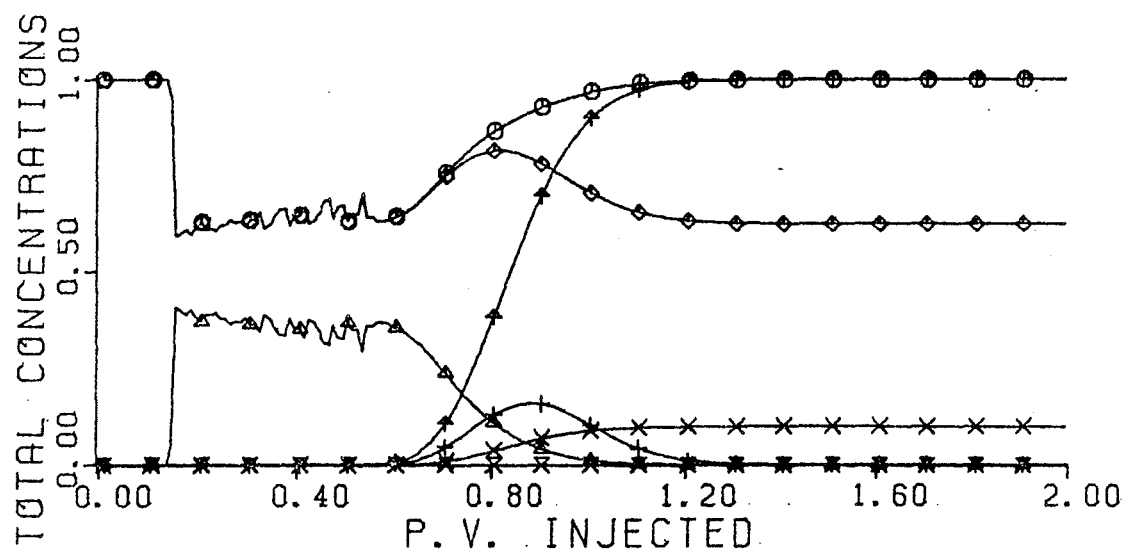


Figure 6.12. History of total concentration in effluent for Run 204
(FDE method, $\Delta t_D = 0.003$, Type II(-), no adsorption).

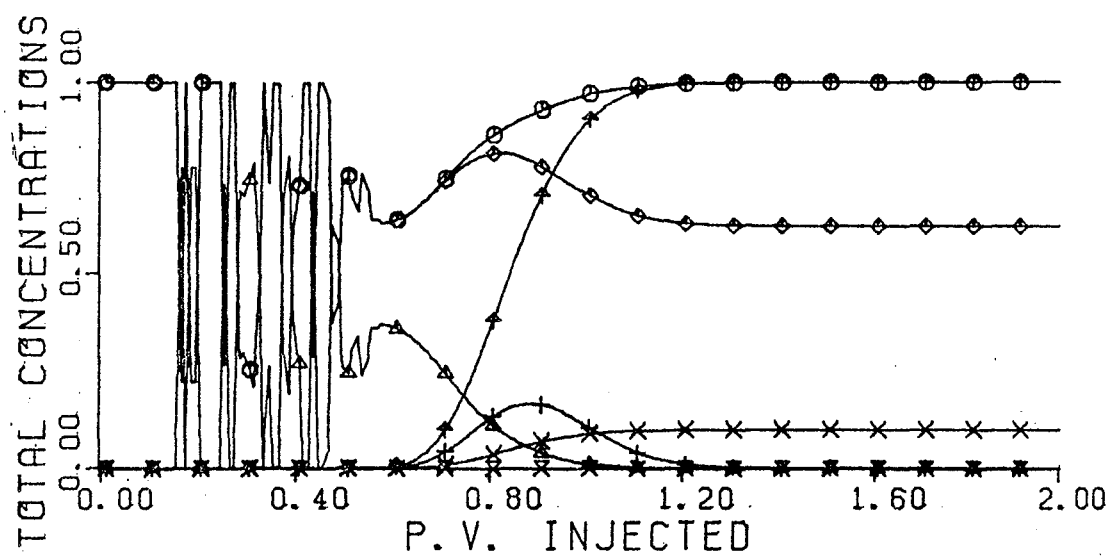


Figure 6.13. History of total concentration in effluent for Run 206
(FDE method, $\Delta t_D = 0.004$, Type II(-), no adsorption).

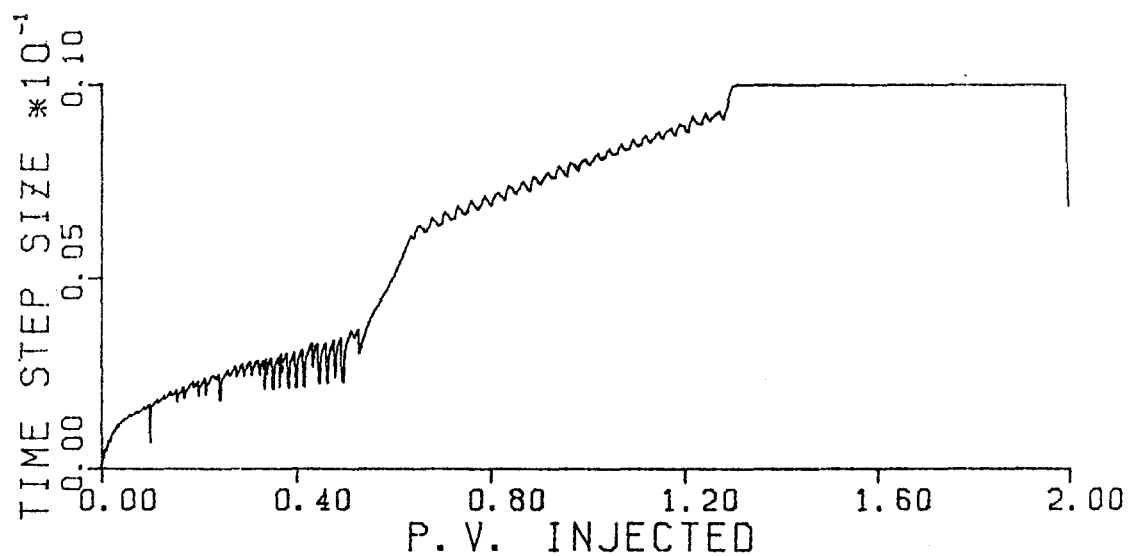
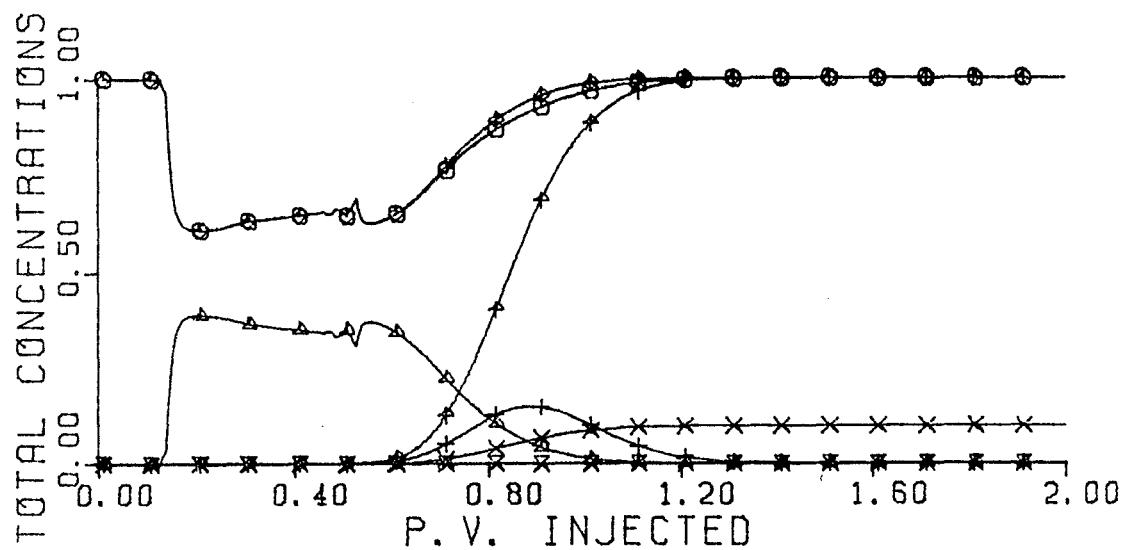


Figure 6.14. Histories of total concentration in effluent and time step size for Run 196 (RK1(2), ERR = 0.00005, Type II(-), no adsorption).

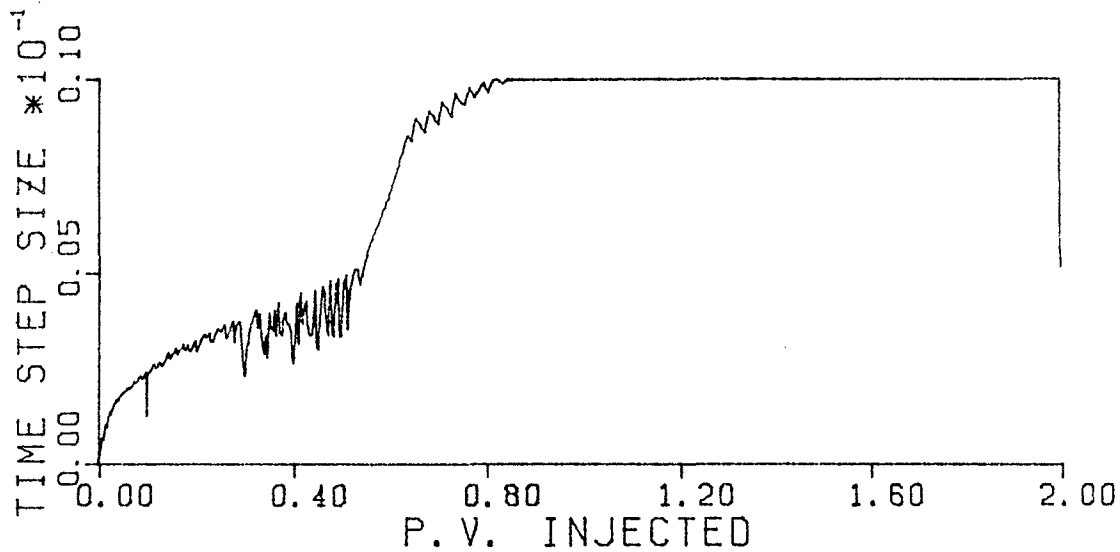
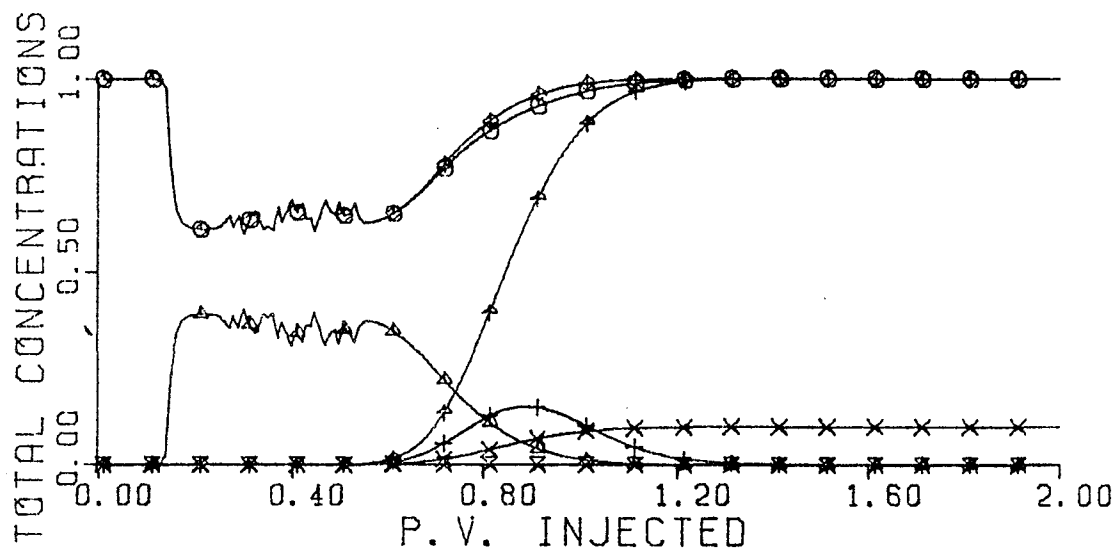


Figure 6.15. Histories of total concentration in effluent and time step size for Run 197 (RK1(2), ERR = 0.0001, Type II(-), no adsorption).

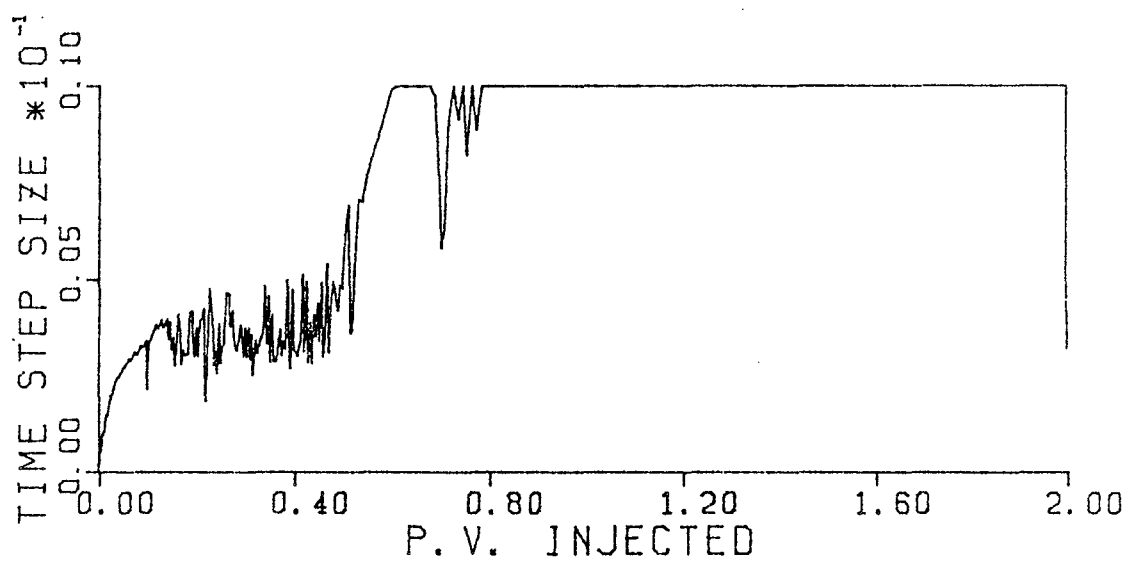
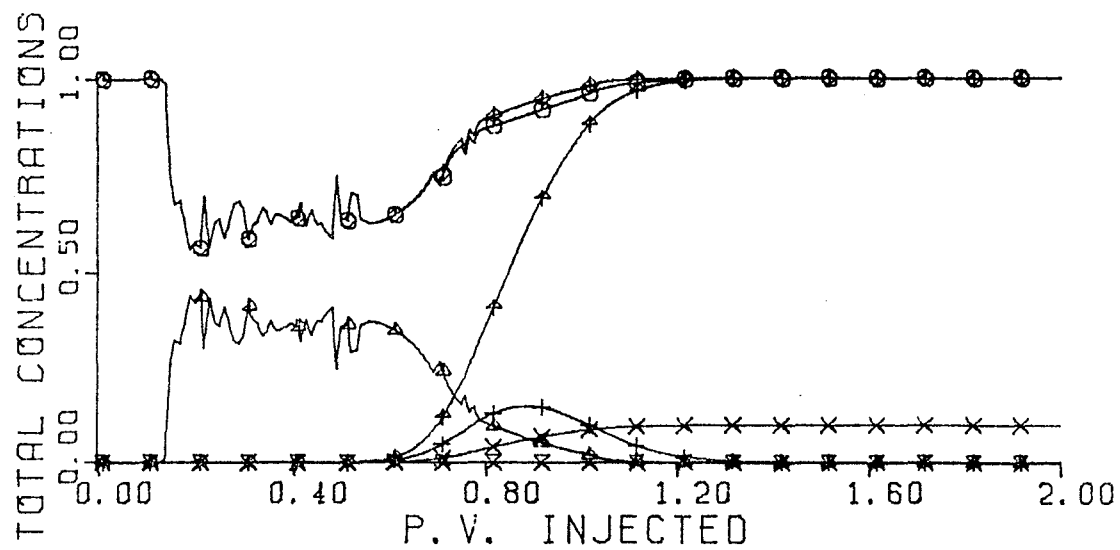


Figure 6.16. Histories of total concentration in effluent and time step size for Run 198 (RK1(2), ERR = 0.002, Type II(-), no adsorption).

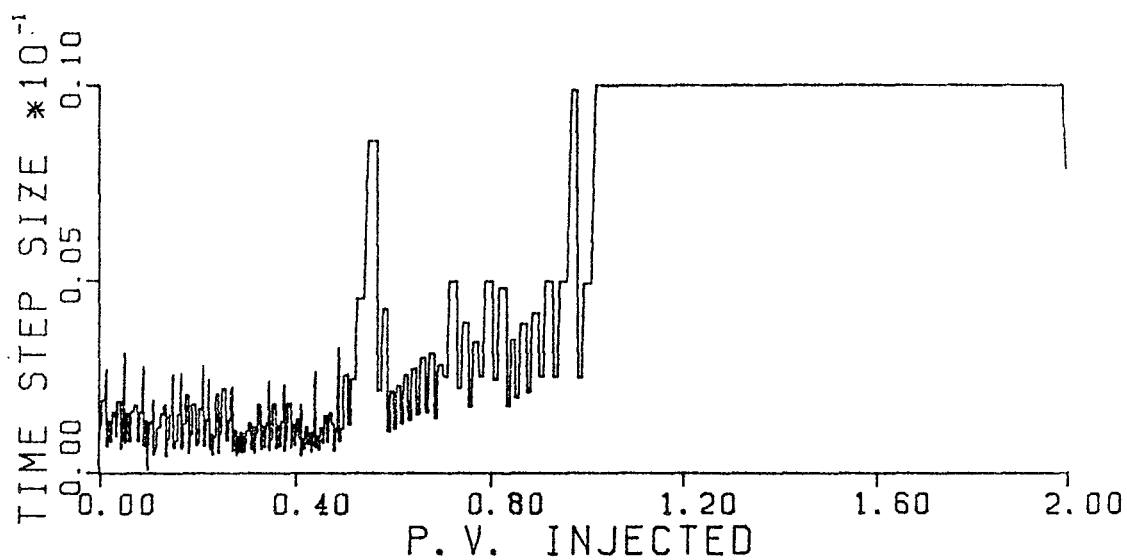
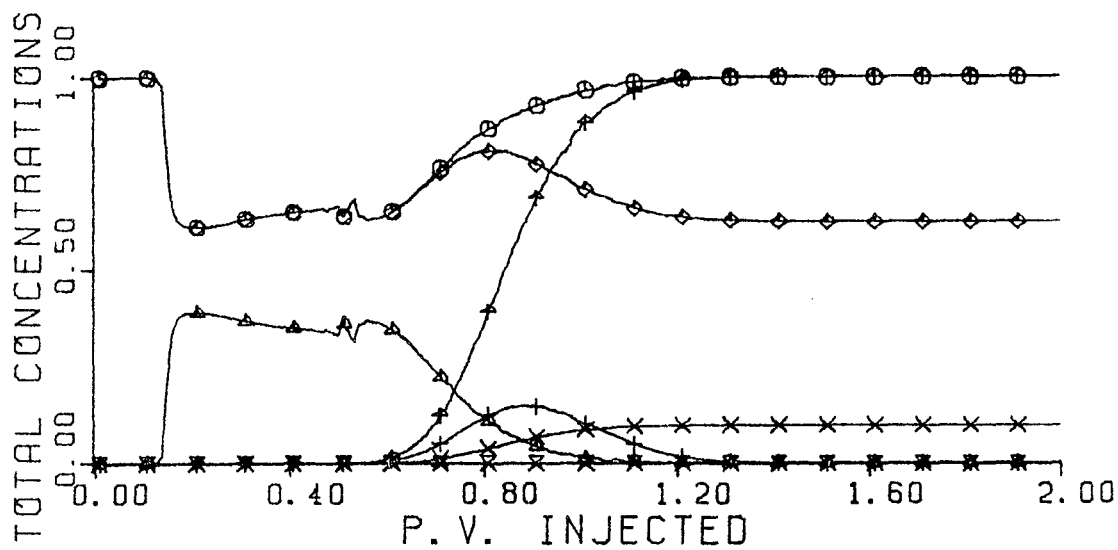


Figure 6.17. Histories of total concentration in effluent and time step size for Run 498 (Adams' method, ERR = 0.0005, Type II(-), no adsorption).

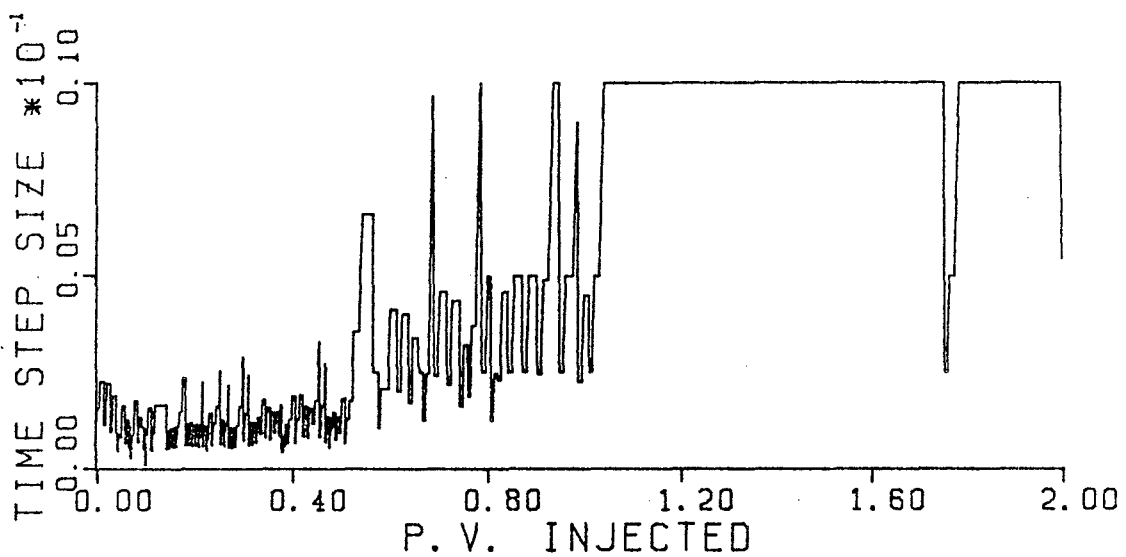
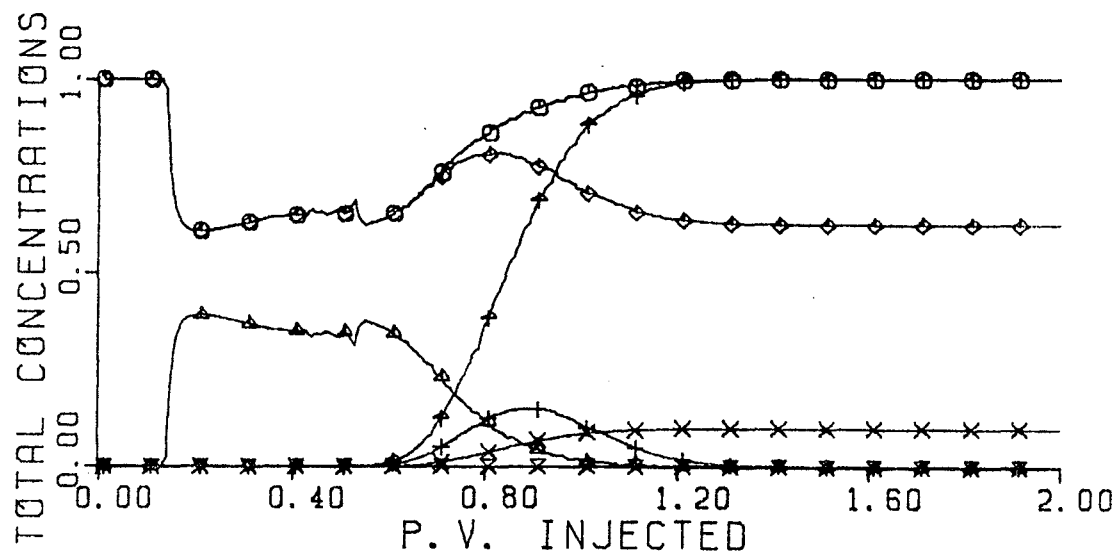


Figure 6.18. Histories of total concentration in effluent and time step size for Run 496 (Adams' method, ERR = 0.001, Type II(-), no adsorption).

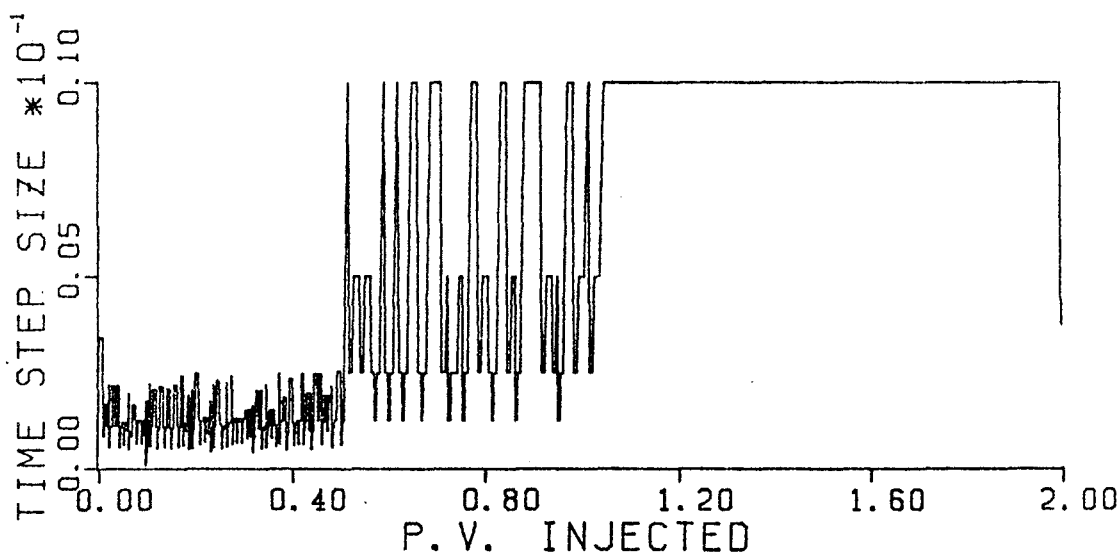
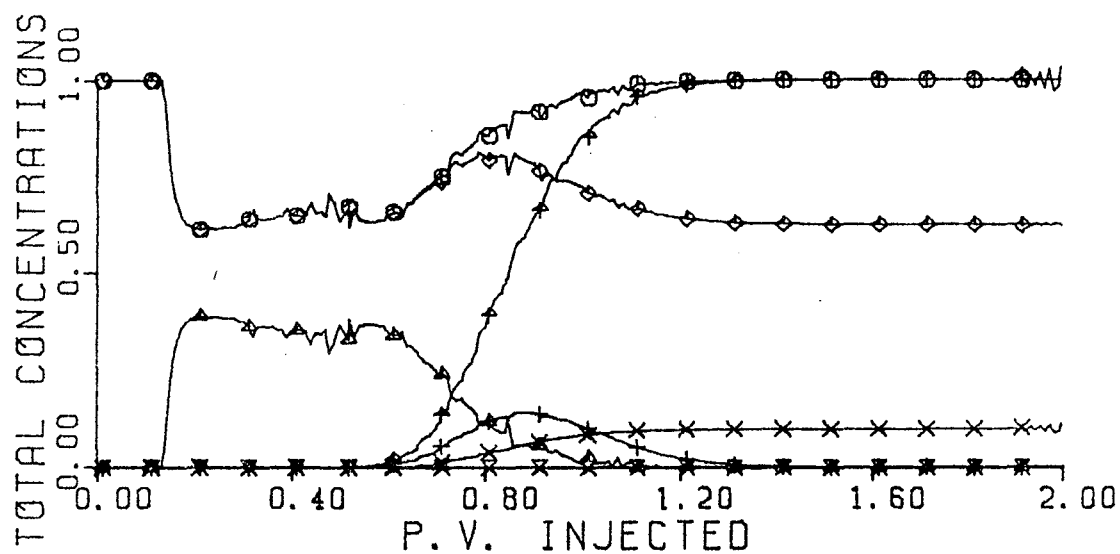


Figure 6.19. Histories of total concentration in effluent and time step size for Run 495 (Adams' method, ERR = 0.01, Type II(-), no adsorption).

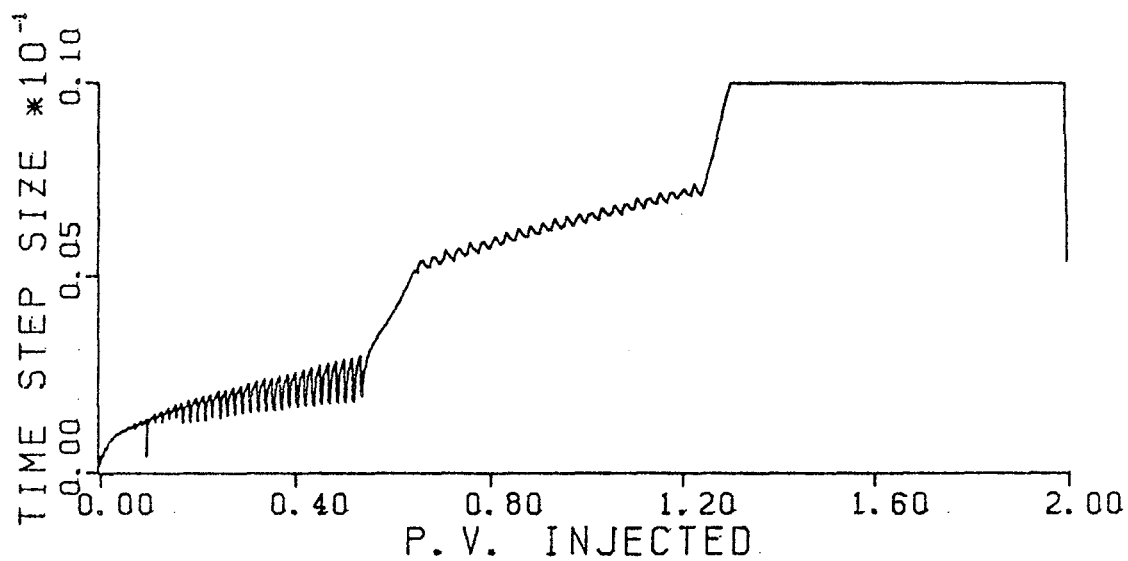
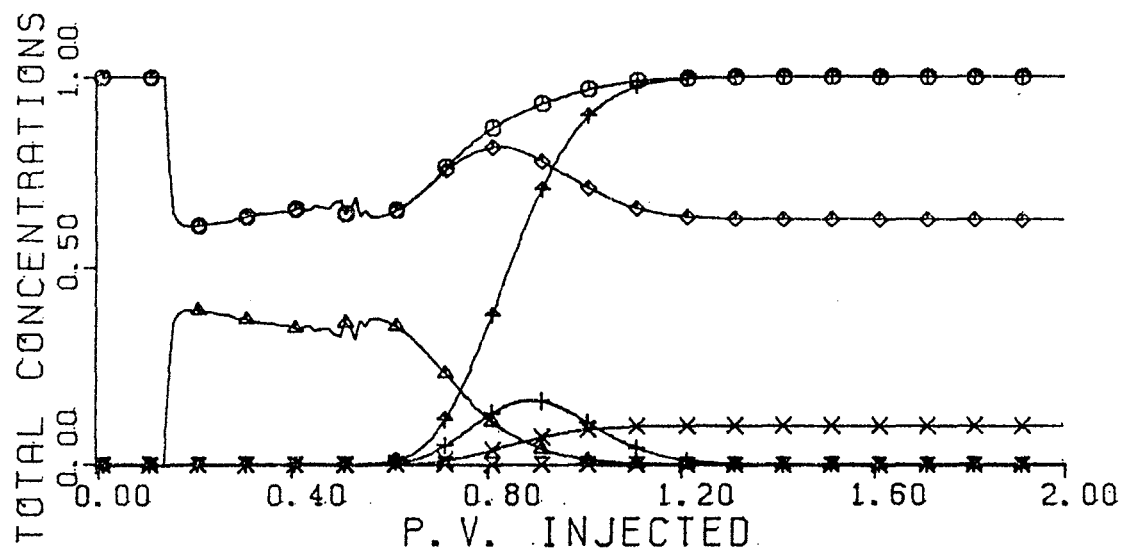


Figure 6.20. Histories of total concentration in effluent and time step size for Run 200 (RK1, ERR = 0.01, Type II(-), no adsorption).

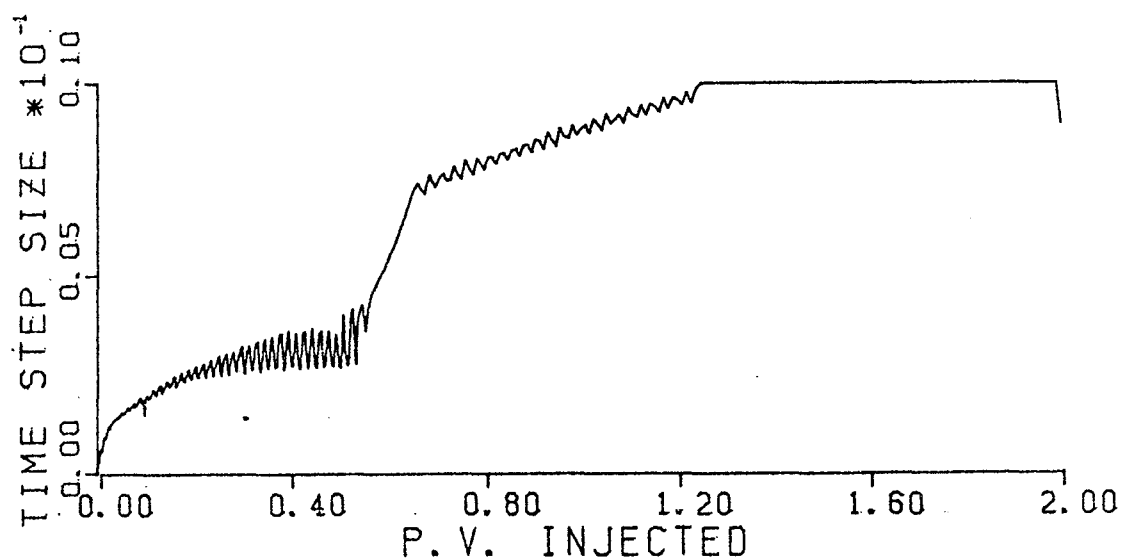
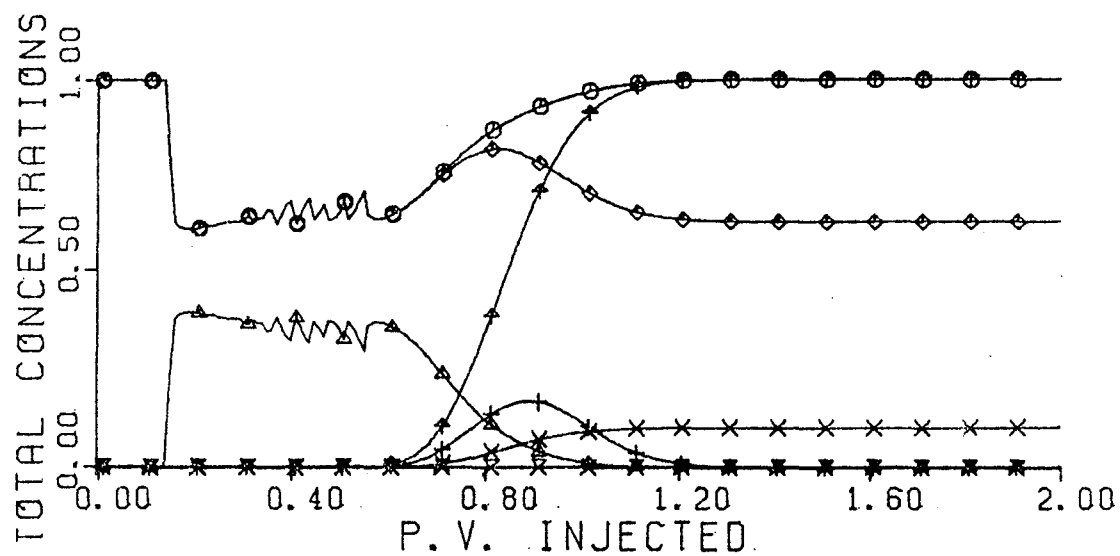


Figure 6.21. Histories of total concentration in effluent and time step size for Run 201 (RK1, ERR = 0.02, Type II(-), no adsorption).

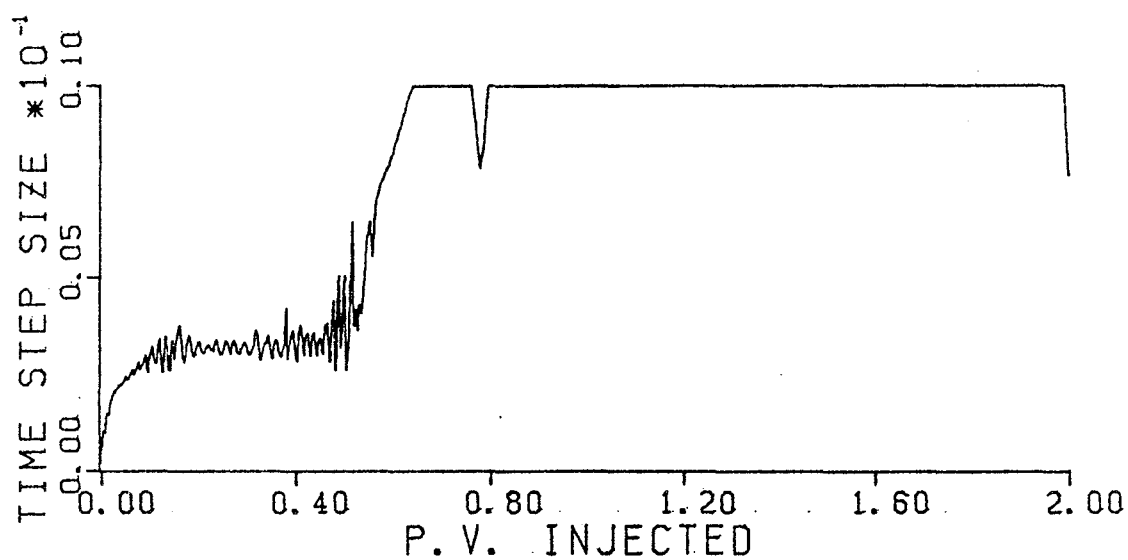
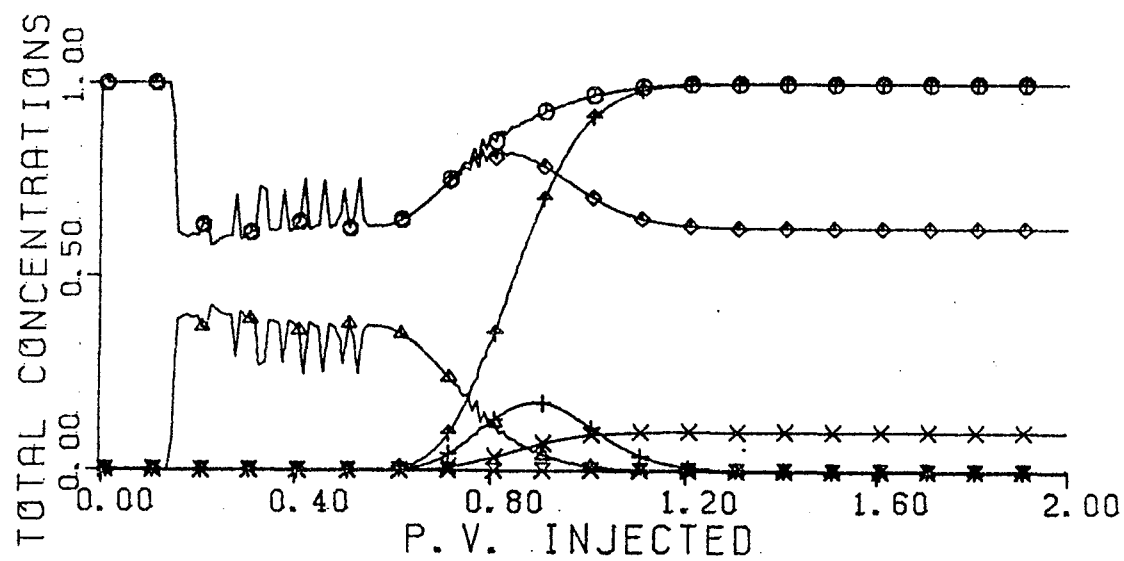


Figure 6.22. Histories of total concentration in effluent and time step size for Run 202 (RK1, ERR = 0.05, Type II(-), no adsorption).

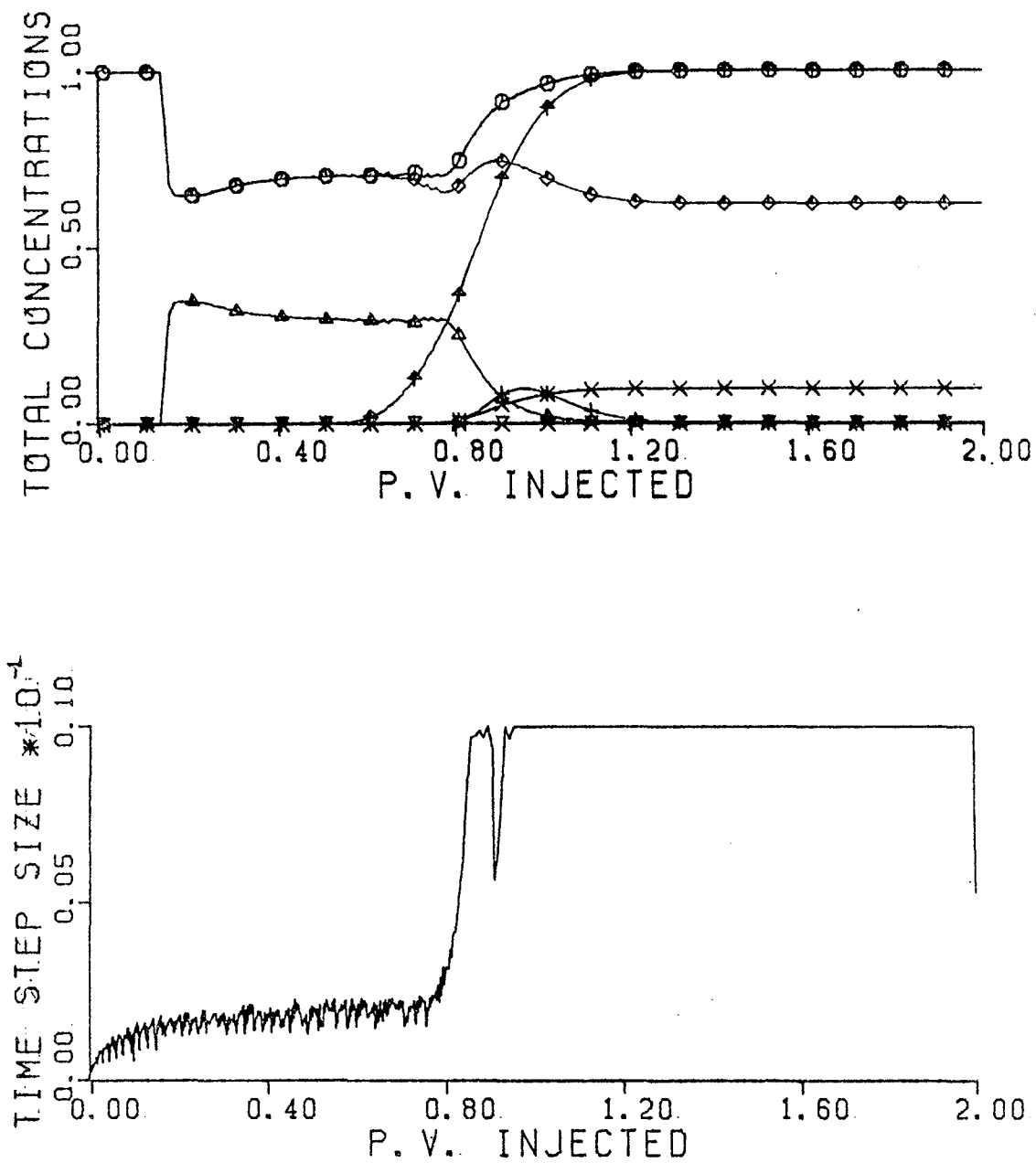


Figure 6.23. Histories of total concentration in effluent and time step size for Run 333 (RK1, ERR = 0.01, Type II(-), with adsorption).

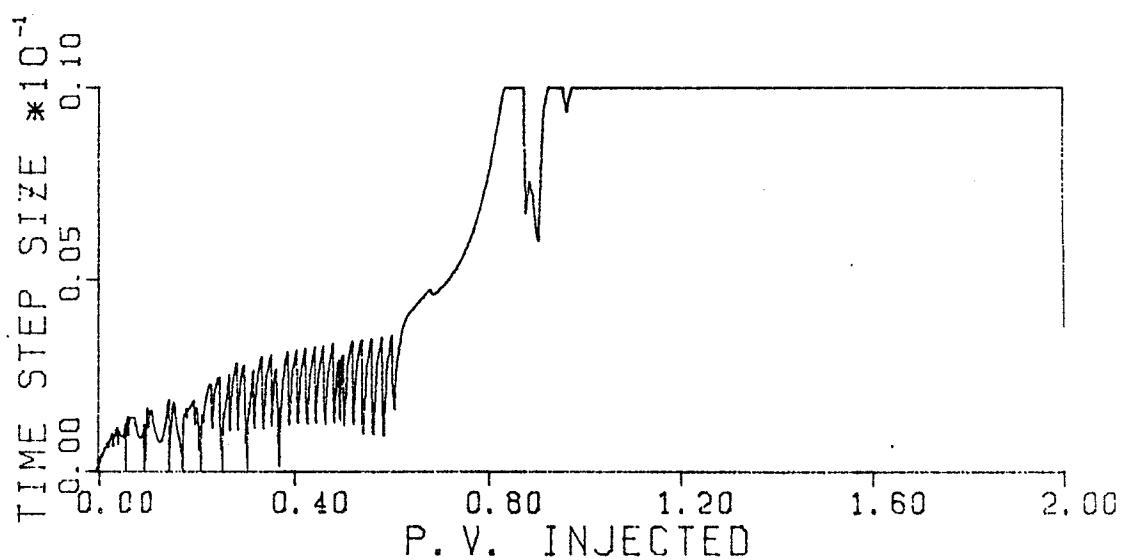
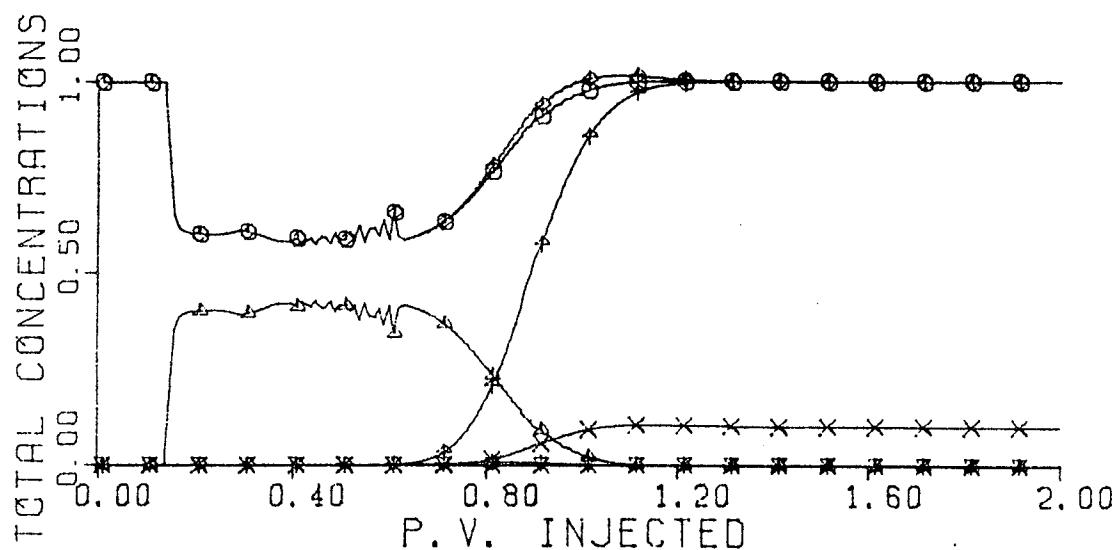


Figure 6.24. Histories of total concentration in effluent and time step size for Run 411 (RK1, ERR = 0.01, Type II(+), no adsorption).

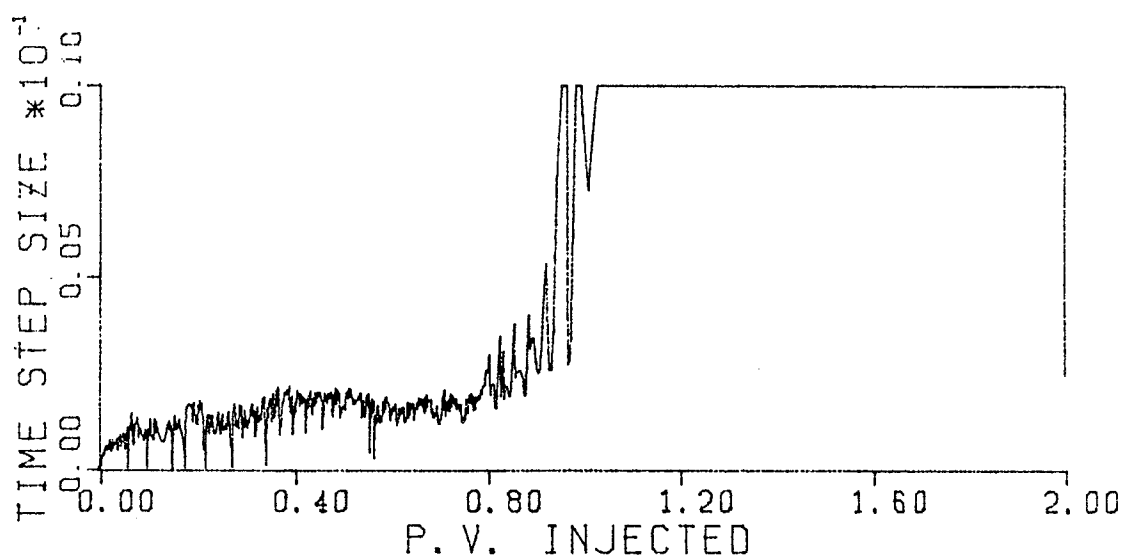
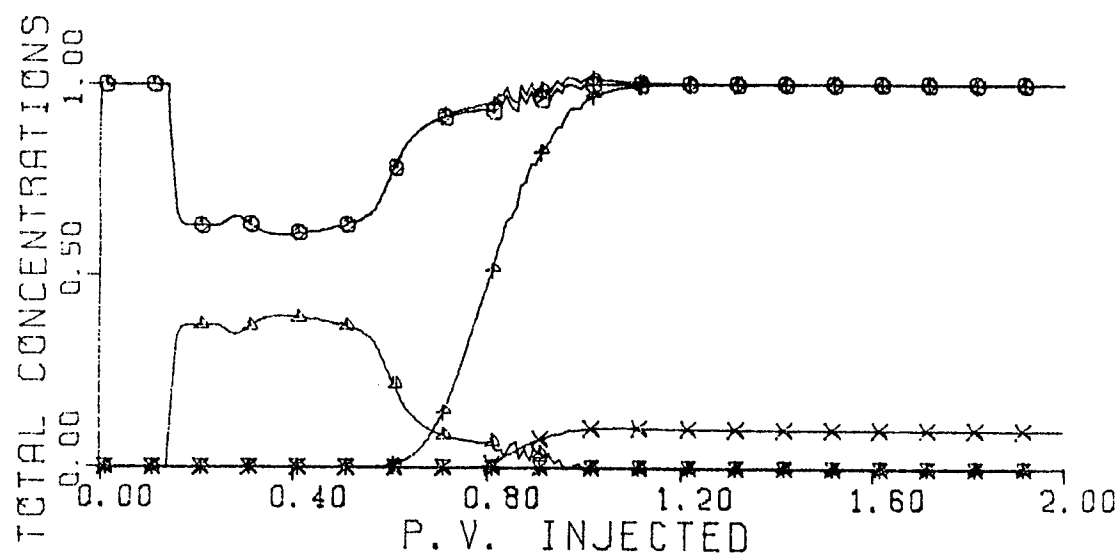


Figure 6.25. Histories of total concentration in effluent and time step size for Run 395 (RK1, ERR = 0.01, Type II(+), with adsorption).

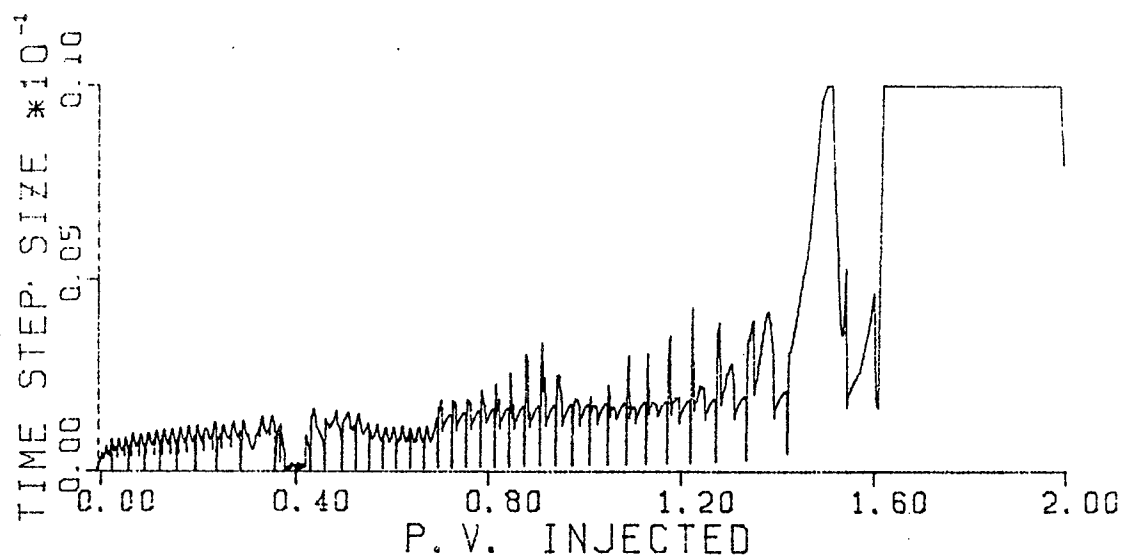
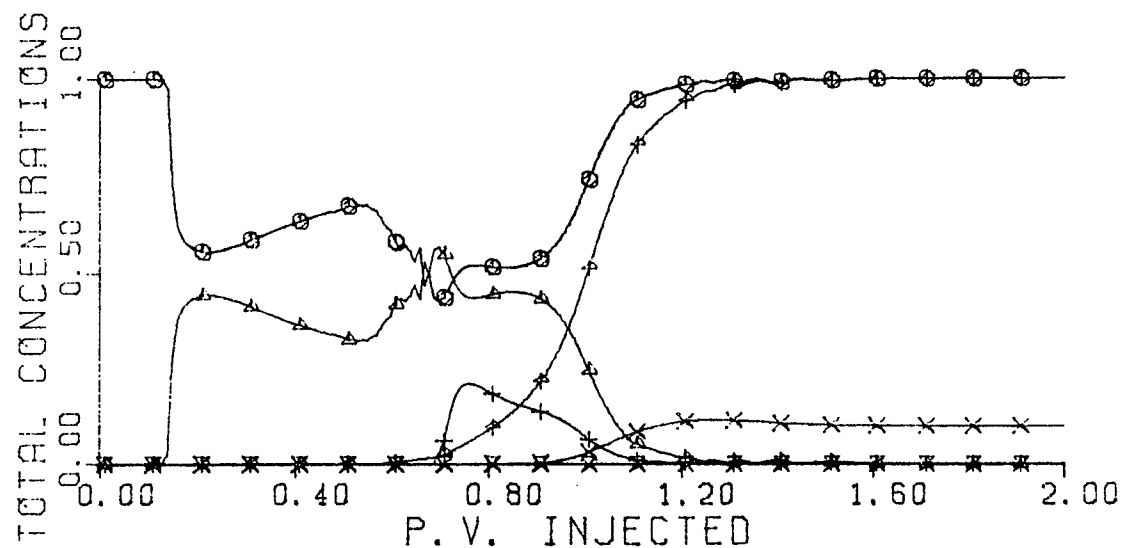


Figure 6.26. Histories of total concentration in effluent and time step size for Run 442 (RK1, ERR = 0.01, Type III, no adsorption).

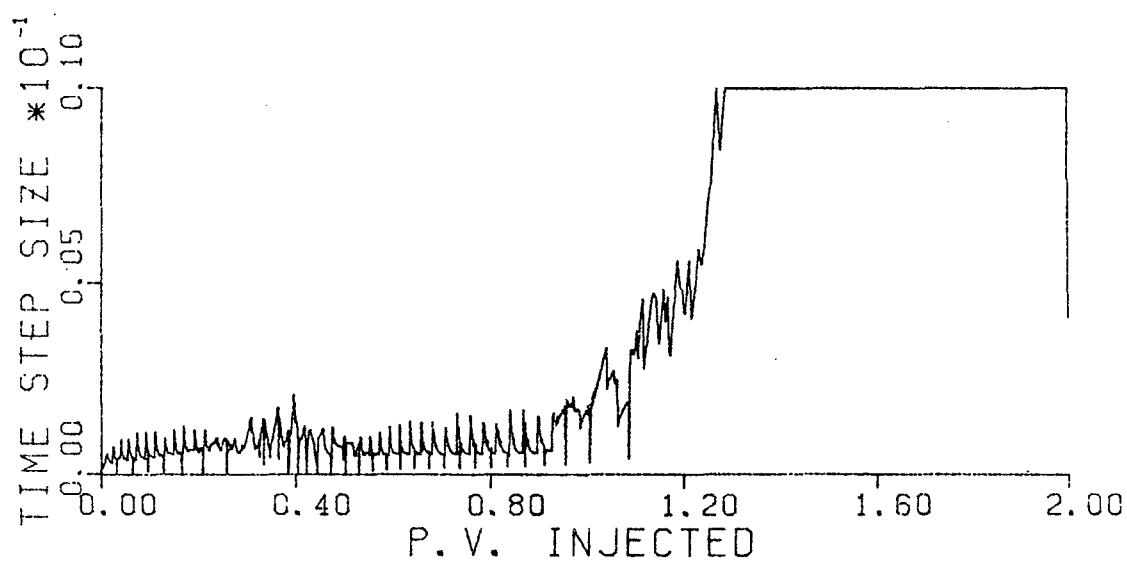
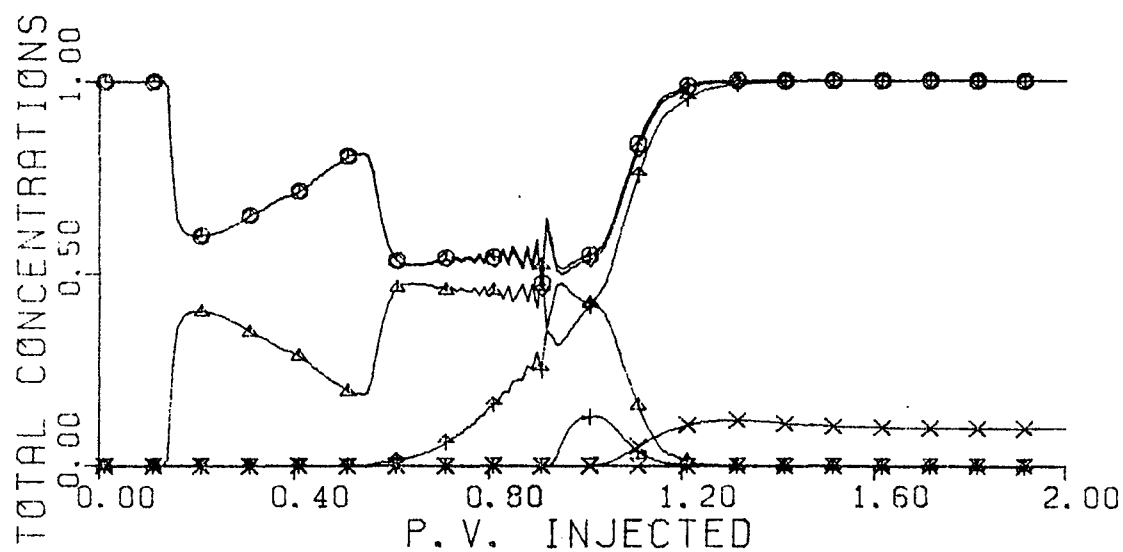


Figure 6.27. Histories of total concentration in effluent and time step size for Run 463 (RK1, ERR = 0.01, Type III, with adsorption).

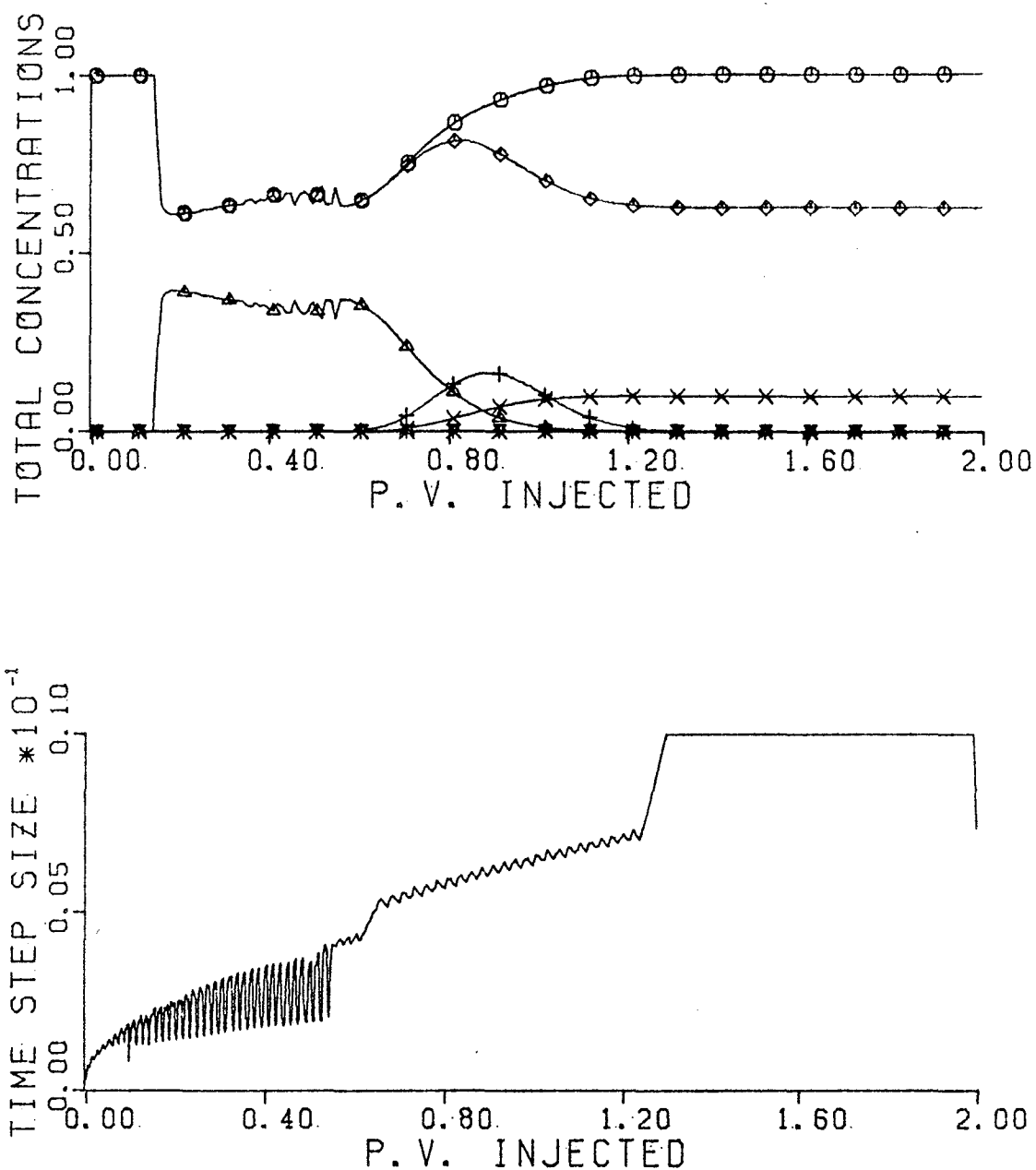


Figure 6.28. Histories of total concentration in effluent and time step size for Run 265 (RK1, ERR = 0.01, Type II(-), no adsorption, no tracer).

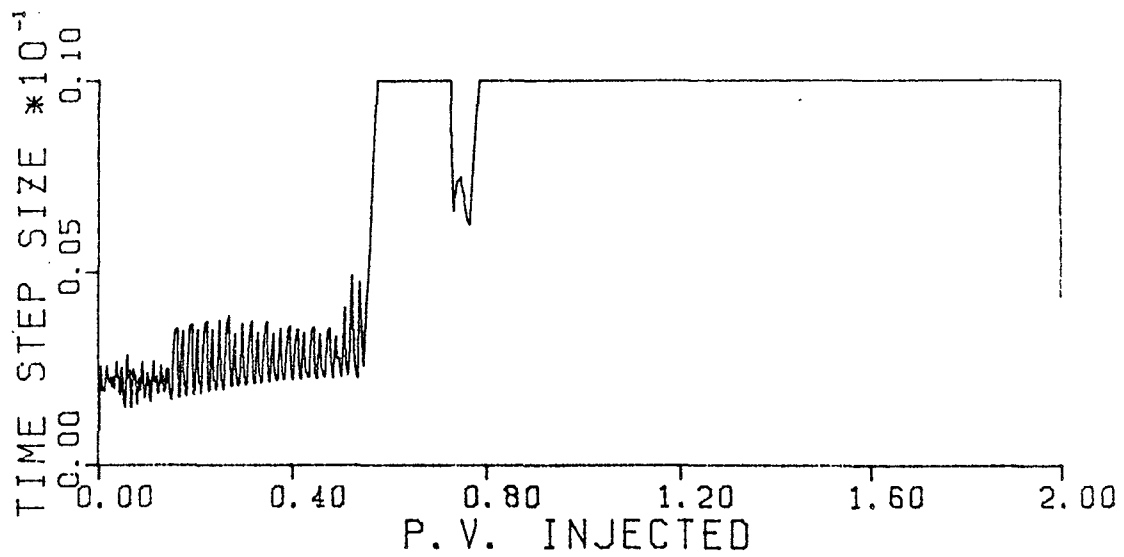
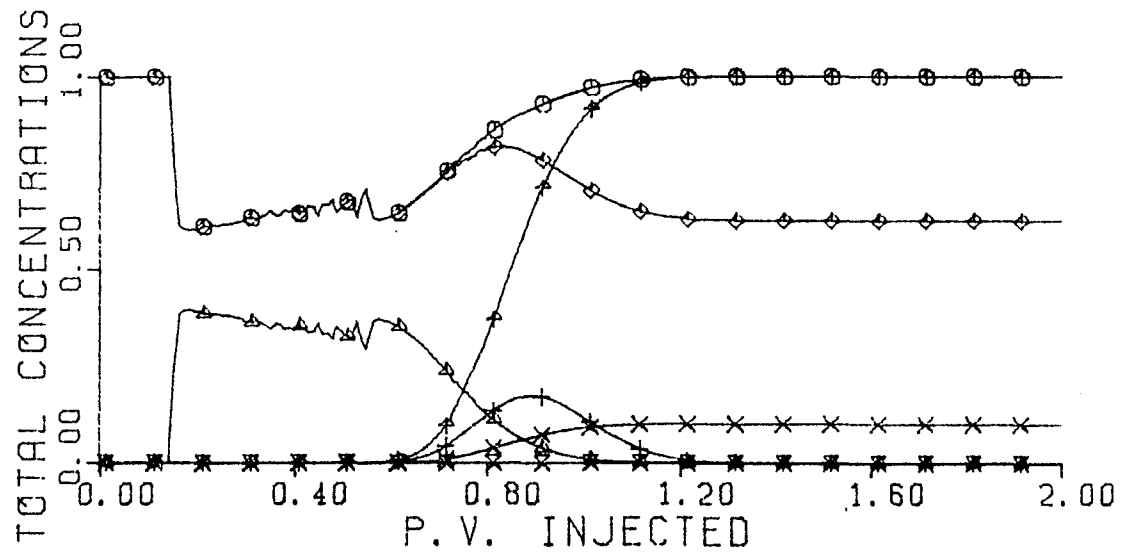


Figure 6.29. Histories of total concentration in effluent and time step size for Run 266 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_1).

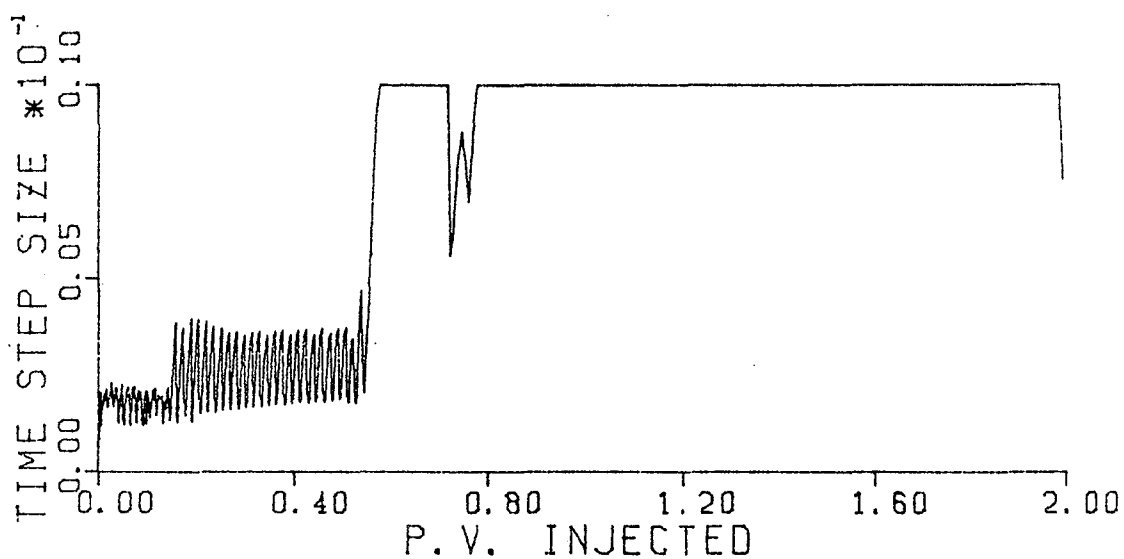
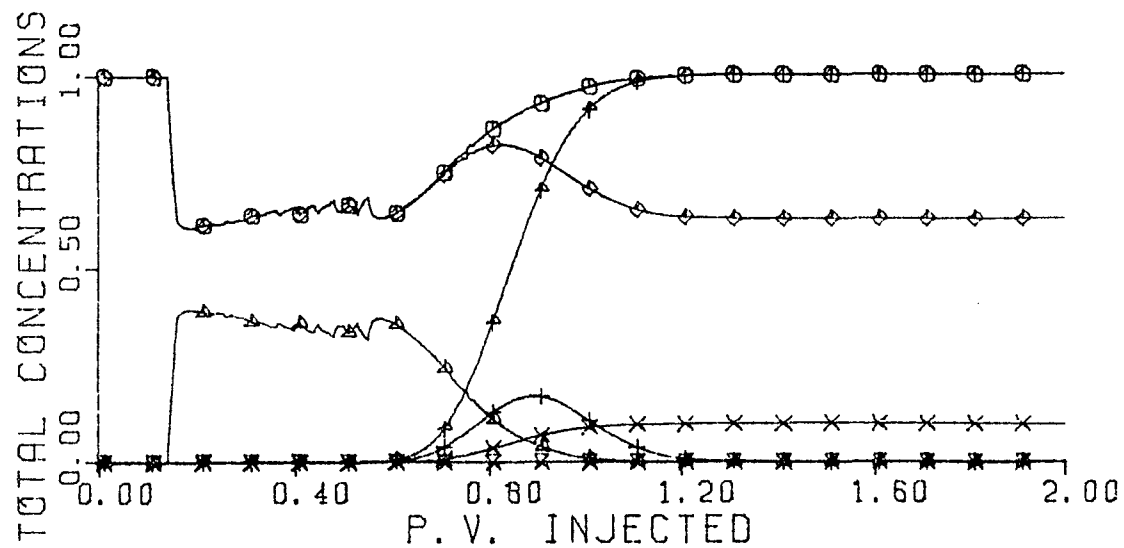


Figure 6.30. Histories of total concentration in effluent and time step size for Run 267 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_2).

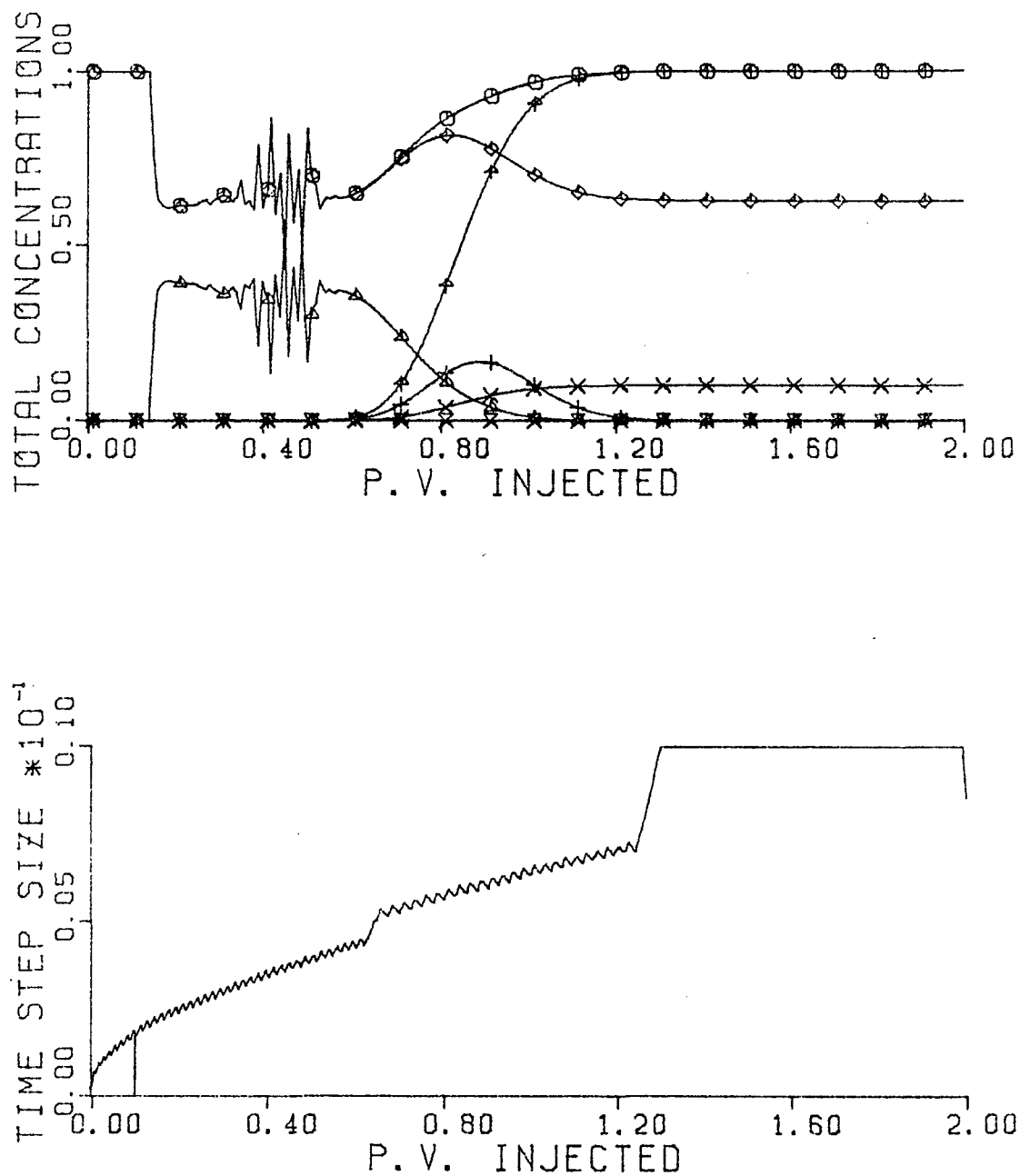


Figure 6.31. Histories of total concentration in effluent and time step size for Run 268 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C₃).

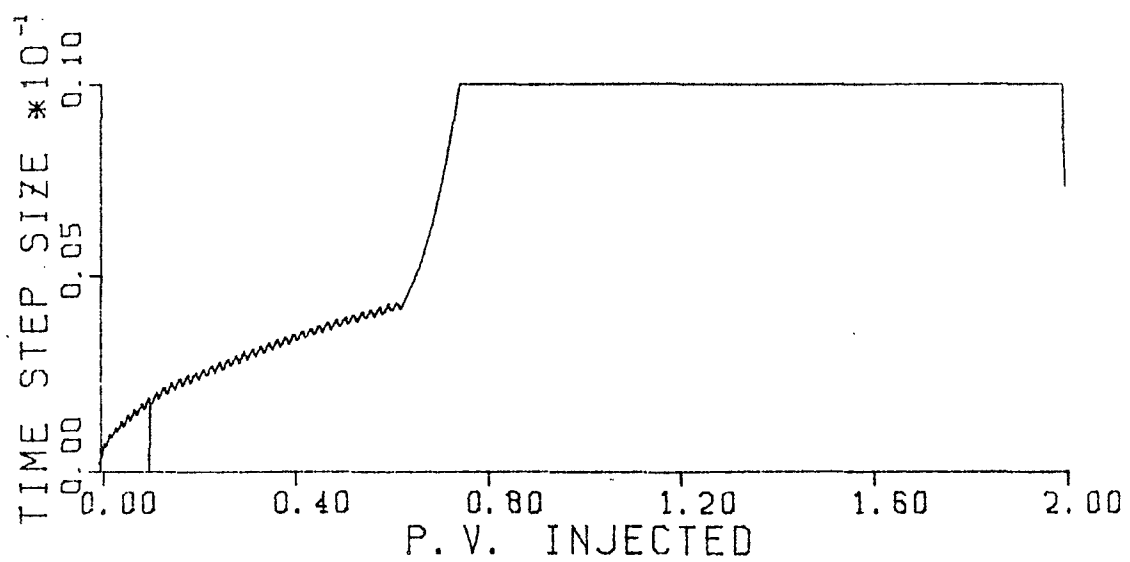
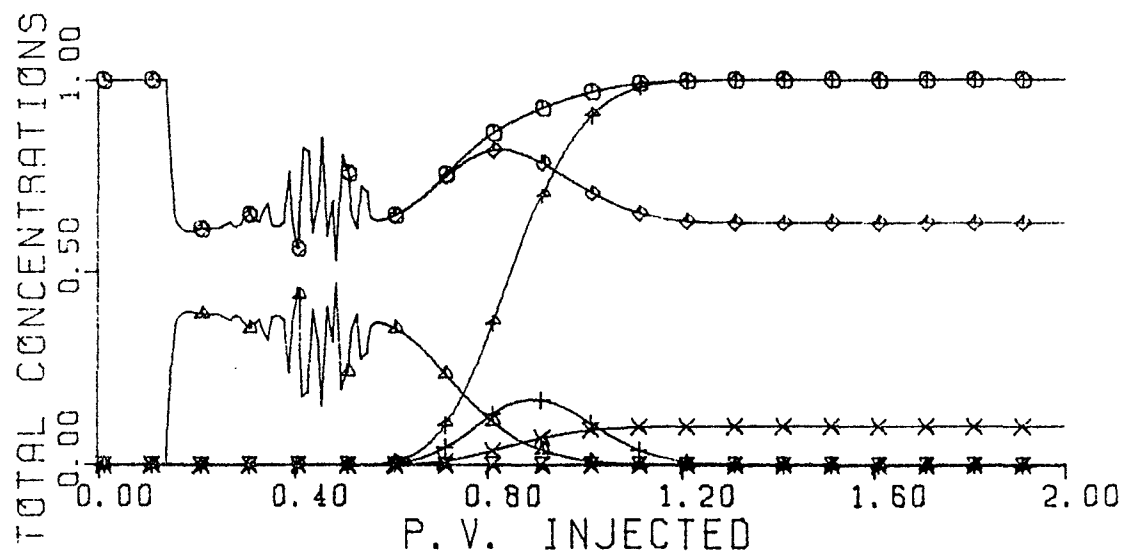


Figure 6.32. Histories of total concentration in effluent and time step size for Run 269 (RK1, ERR = 0.01, Type II(-), no adsorption, error checked only for C_4).

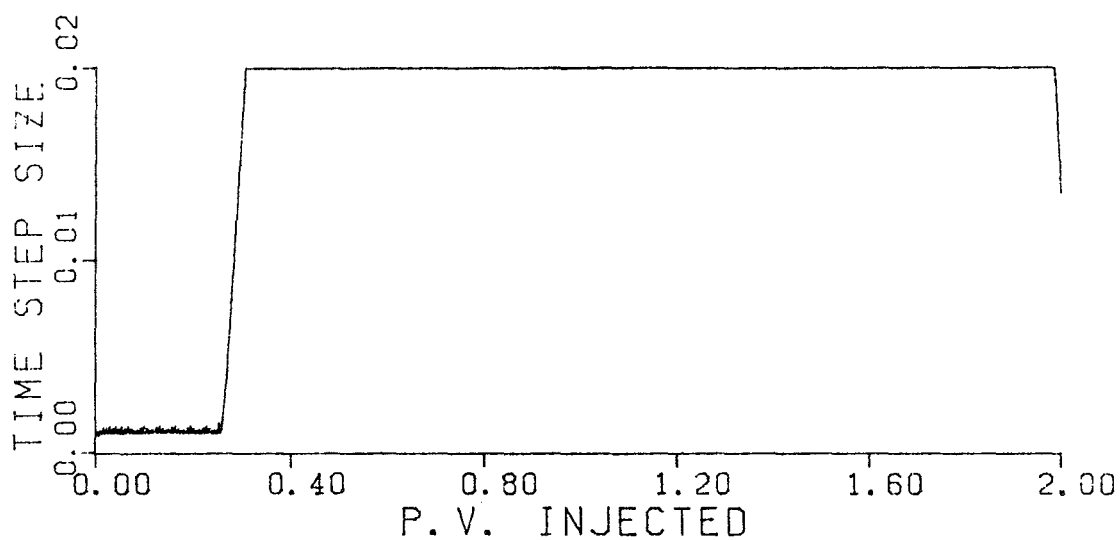
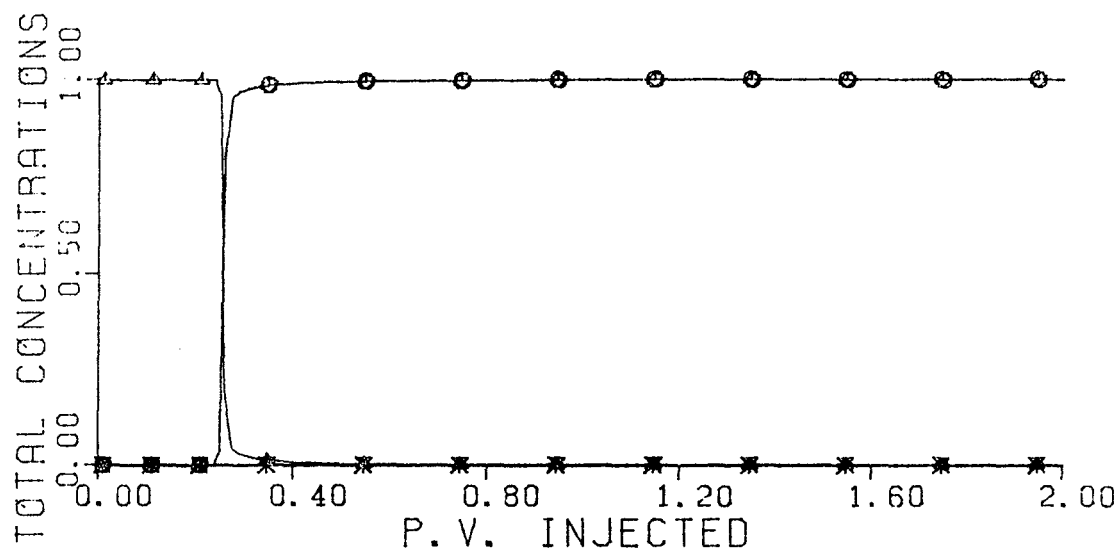


Figure 6.33. Histories of total concentration in effluent and time step size for Run 229 (RK1, ERR = 0.01, waterflooding, no tracer).

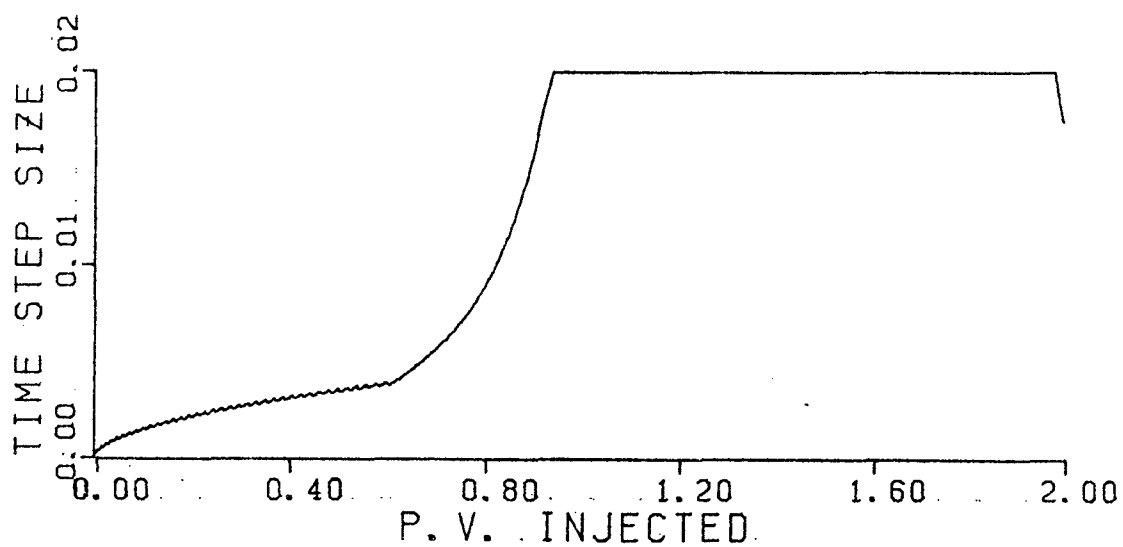
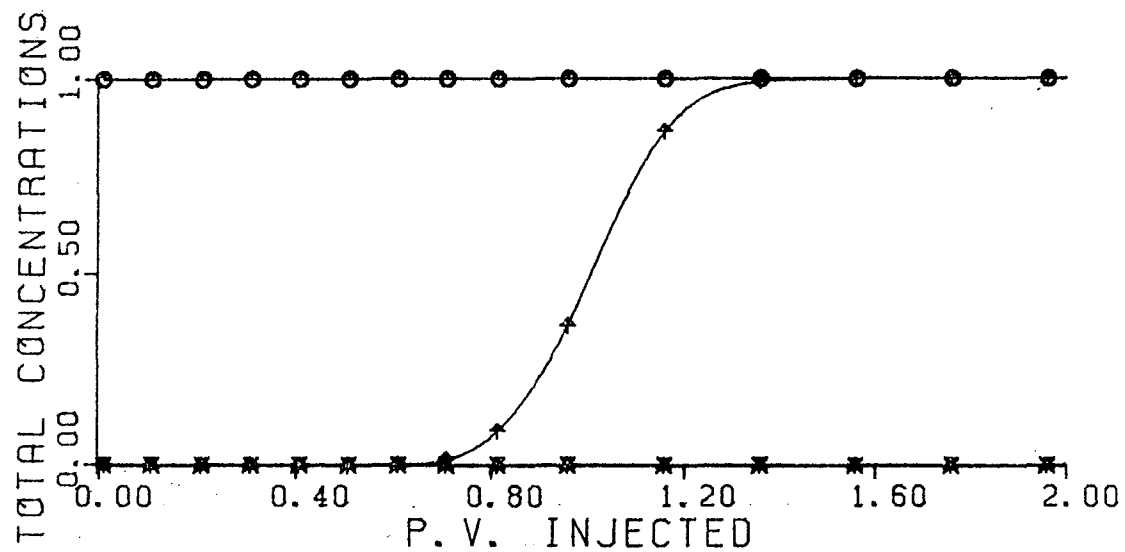


Figure 6.34. Histories of total concentration in effluent and time step size for Run 260 (RK1, ERR = 0.01, miscible displacement with tracer).

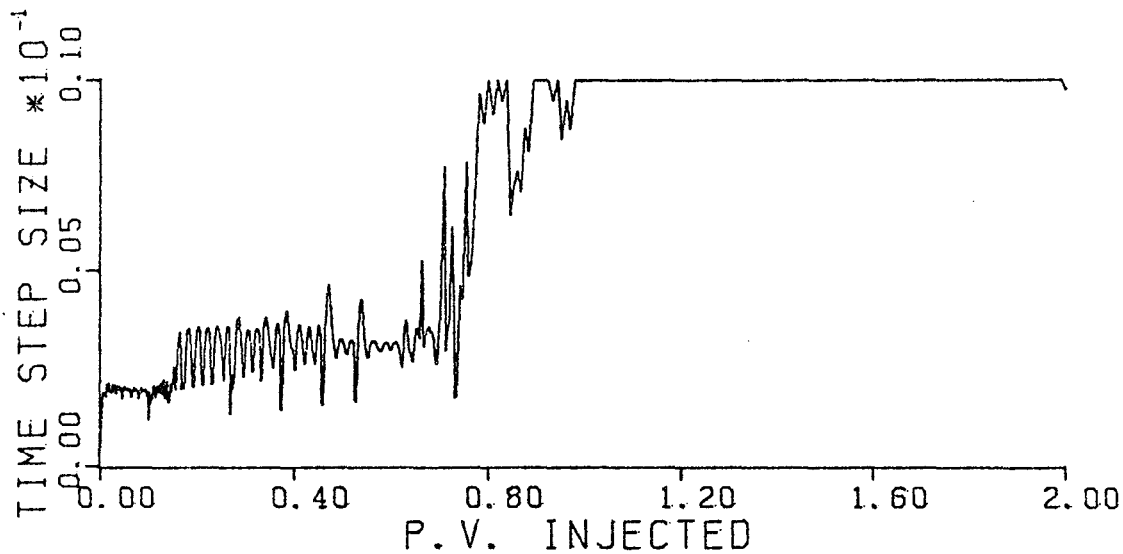
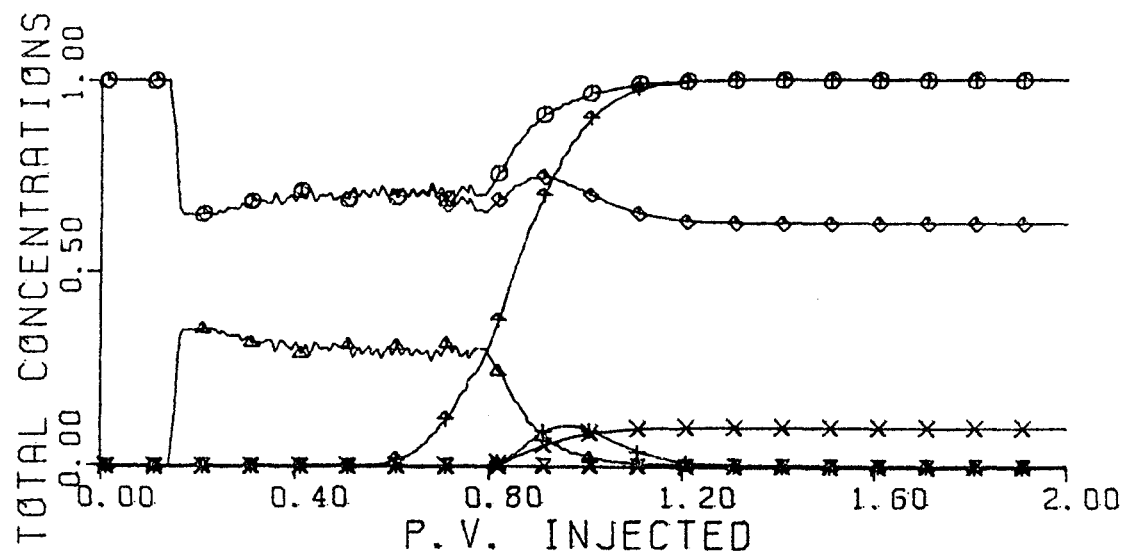


Figure 6.35. Histories of total concentration in effluent and time step size for Run 522 (RK1, ERR = 0.01, Type II(-), with adsorption, error checked only for C_2).

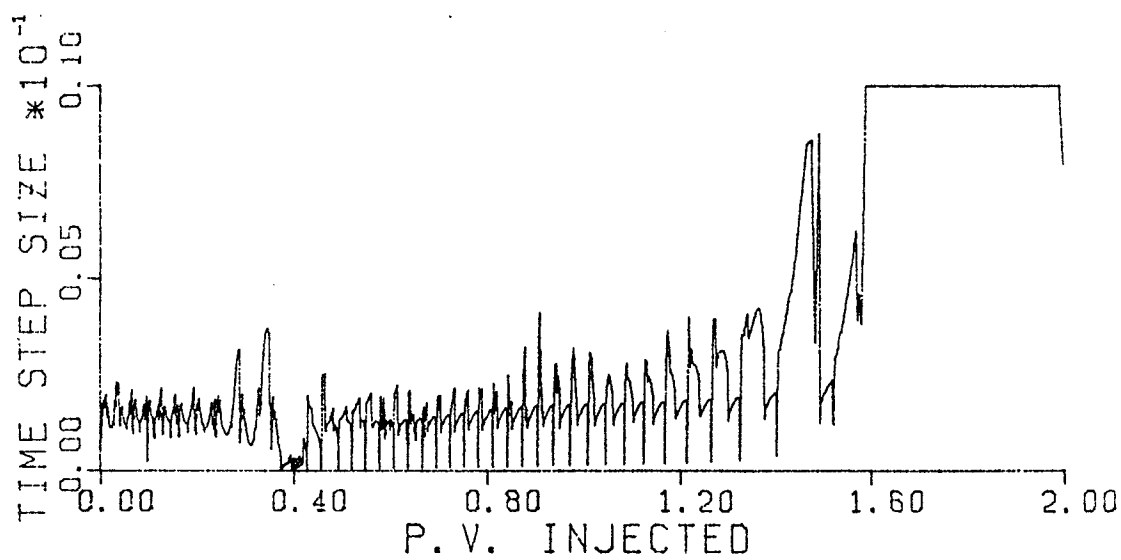
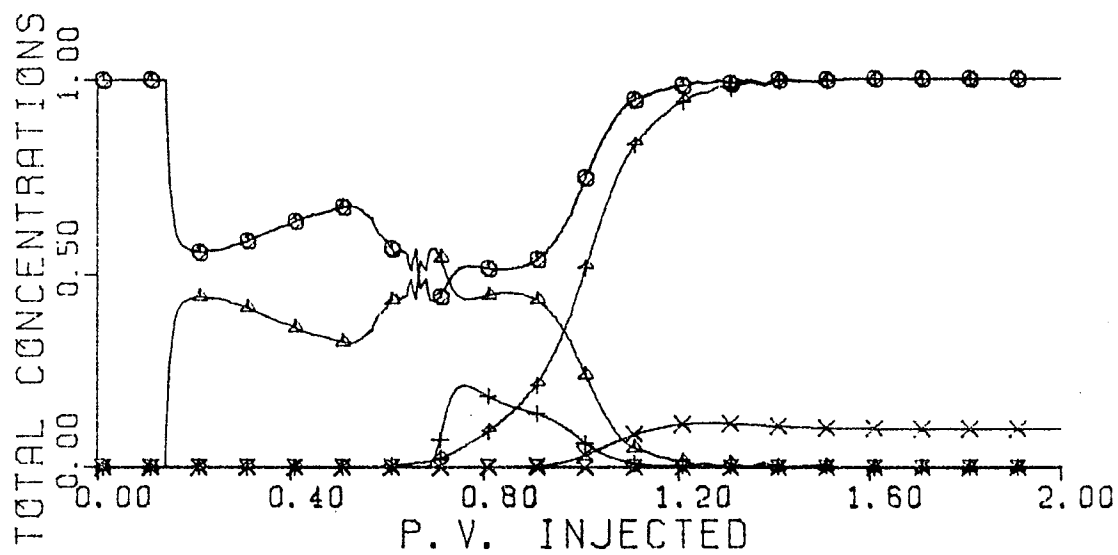


Figure 6.36. Histories of total concentration in effluent and time step size for Run 446 (RK1, ERR = 0.01, Type III, no adsorption, error checked only for C_2).

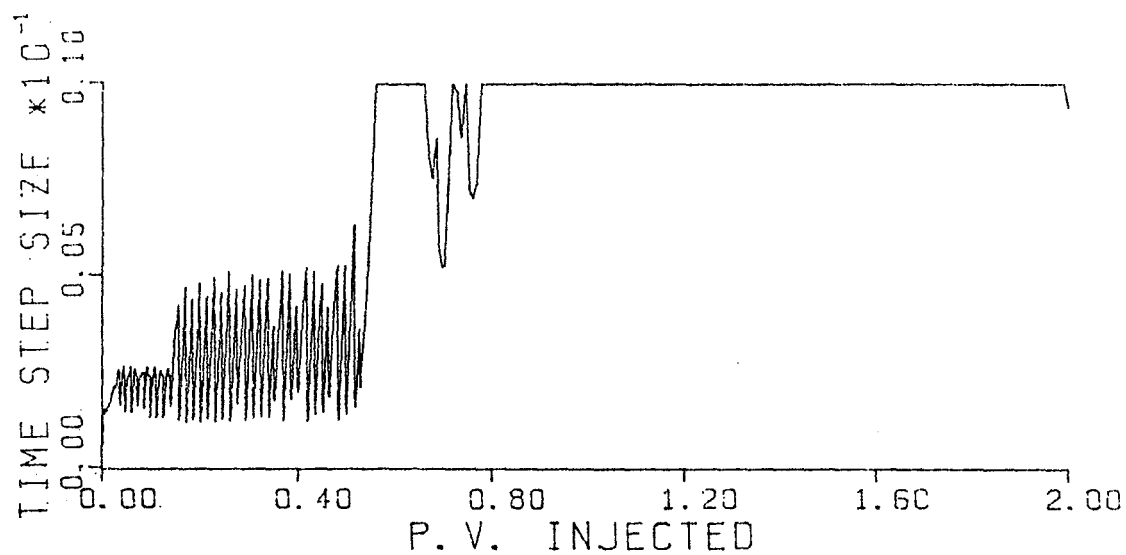
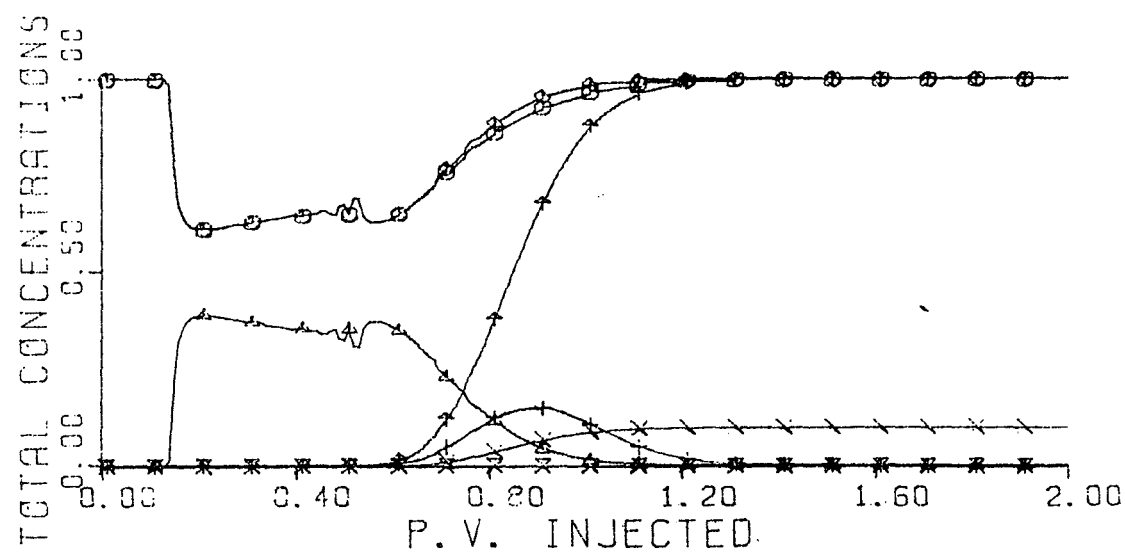


Figure 6.37. Histories of total concentration in effluent and time step size for Run 629 (RK1(2), ERR = 0.0001, PCT = 0.25, YBIAS = 1.0, Type II(-), no adsorption).

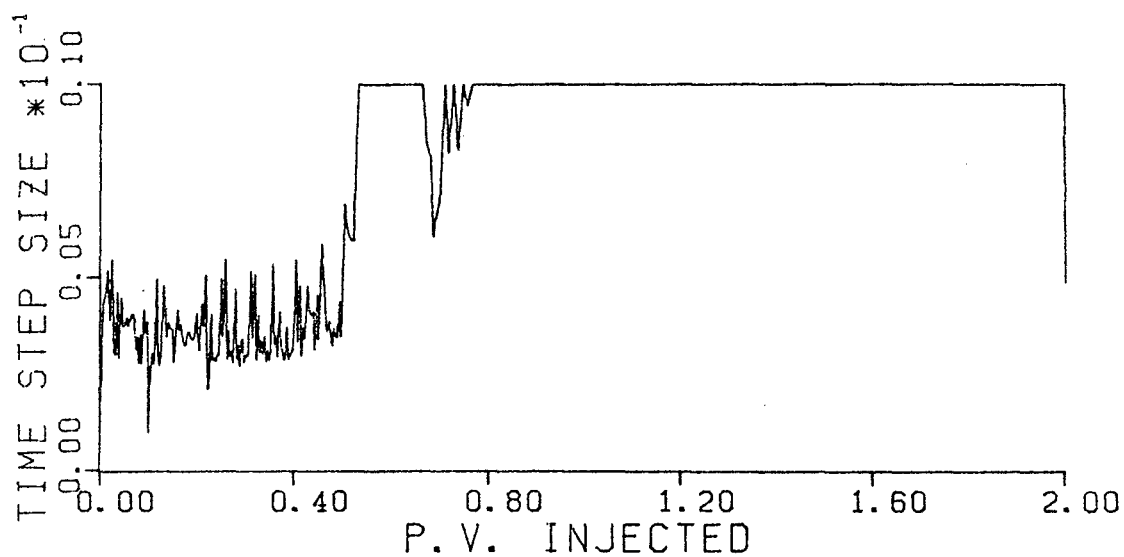
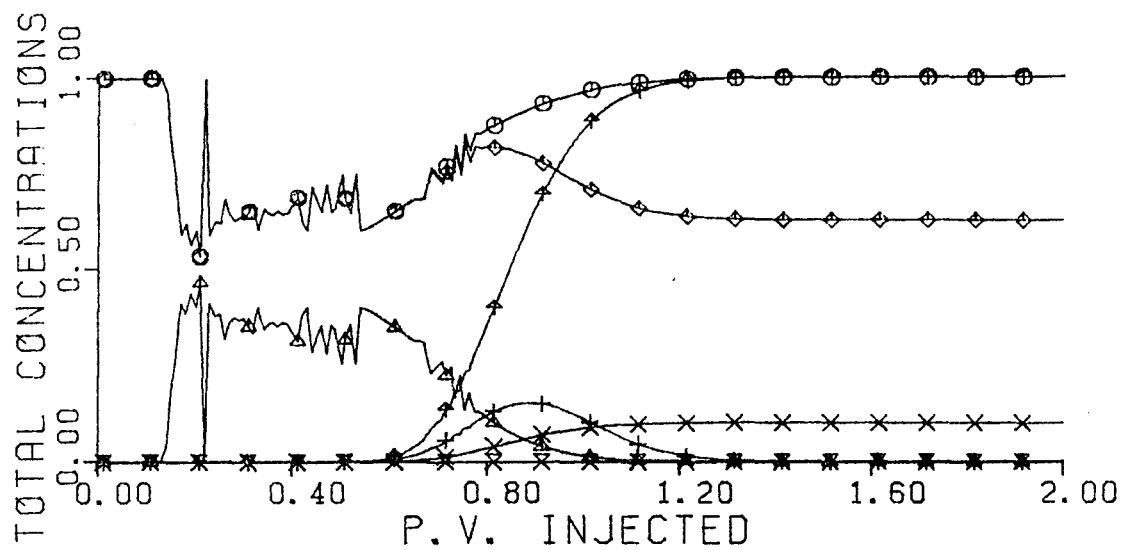


Figure 6.38. Histories of total concentration in effluent and time step size for Run 647 (RK1(2), ERR = 0.0001, PCT = 0.50, YBIAS = 1.0, Type II(-), no adsorption).

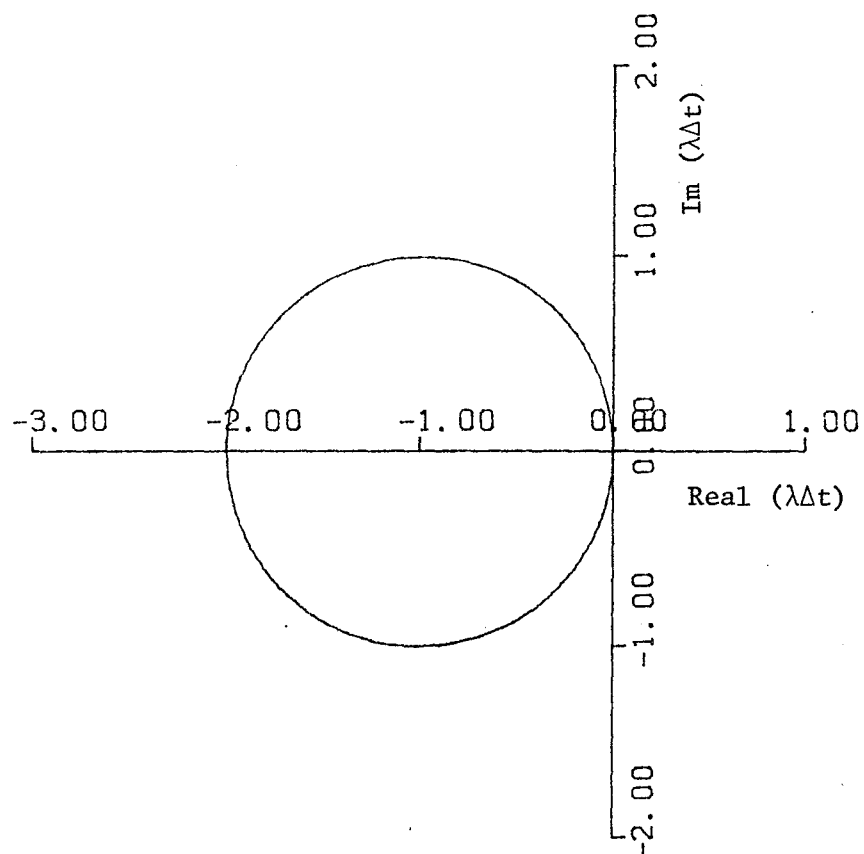


Figure 6.39. Stability region for RK1 or Forward Euler.

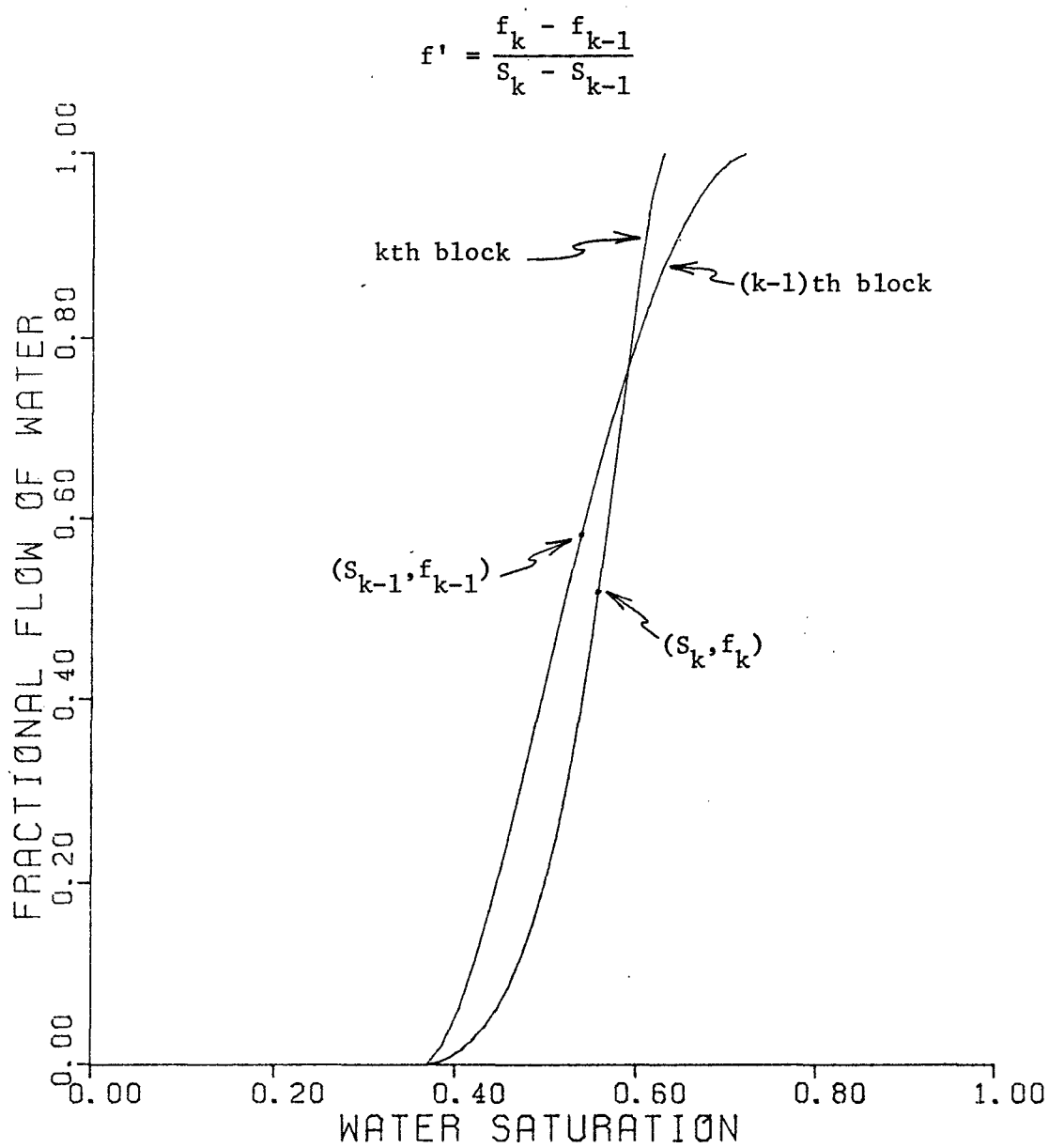


Figure 6.40. Explanation of negative value for f' .

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY

Some comparisons were made among three different three-phase flow models. When the salinity was constant, the difference in oil recovery and surfactant trapping among the models was rather large, especially if the injected amount of surfactant was small. With a salinity gradient, there was only a small difference in both oil recovery and surfactant trapping. Since in all models surfactant trapping was significant and since it is highly uncertain, this is still another important reason for designing a micellar flood with a salinity gradient.

Semi-discrete methods with various ODE integrators were implemented in a 1-D micellar/polymer flooding simulator. The ODE integrators used are 1) Runge-Kutta method: RK1; 2) Runge-Kutta-Fehlberg method: RK1(2); 3) Adams' methods; 4) Backward differentiation (Gear's) methods. The first two methods are explicit methods while the last two are implicit with a predictor-corrector algorithm.

With respect to computation time, RK1 was the best among the ODE integrators used. Compared with the fully-discrete with forward time Euler (FDE) method, RK1 may save 20 to 30% or more computation time, although it depends on phase behavior, adsorption, total volume injected, etc. For some cases such as Type II(+) phase behavior with adsorption, or Type III phase behavior, FDE required less computation time than RK1. Furthermore, the truncation error associated

with the time integration for RK1 is larger than for other ODE integrators. The degree of numerical dispersion changes as the time step size is varied with RK1. However, with Chaudhari's technique²⁸, it may be possible to eliminate the problem by cancelling the numerical dispersion at every time step.

Although the computation time with RK1(2) was not as good as had been expected, a much smaller error associated with time integration could be achieved without large computation time. RK1(2) should be used when a higher order approximation to the spatial derivatives are applied, or when a finite element technique is introduced to treat the spatial domain, since these techniques yield much smaller error in the spatial domain.

To achieve less computation time with RK1 or RK1(2), two methods are suggested. One is to check the error only in the oil or water component with small YBIAS, and with PCT nearly unity. The other is to make YBIAS equal unity and use smaller PCT.

The semi-discrete methods do have the advantage of automatically selecting a sufficiently small time step to avoid large truncation error and/or instabilities. Since this time step is problem-dependent, one can avoid a costly trial-and-error determination of the required value, or the use of an excessively conservative (small) value. There are of course many problems, conditions, and physical options not tried during this study for which it would be a great advantage.

Predictor-corrector methods such as Adams' methods or backward differentiation (Gear's) methods did not work as well as the two explicit techniques. However, it may be too early to conclude these implicit or semi-implicit methods are not as good. Since a packaged program was used to implement the techniques, the details are not clear, but there may be some improvement possible.

If one desires to test predictor-corrector methods further, fixed lower order methods should be tried. Since we are dealing with partial differential equations, it is of no use to achieve a much higher accuracy in time integration compared with spatial integration. Furthermore, higher order methods have a smaller stability region, which may impose more limitation on the time step size.

There was one problem concerning the step size control employed in DGEAR, which varies time step size based on only relative error, without YBIAS (see equation (5.7)). Since most variables change between zero and unity, it is risky not to use YBIAS, or a combination of relative and absolute error⁶².

The most difficult problem associated with predictor-corrector methods is convergence. Since the equations involved in micellar/polymer flooding are highly non-linear, much investigation and effort may be required. If one employs Newton's iteration, one should take advantage of the sparseness of the Jacobian matrix.

Subprogram DER, which calculates the derivative of each component at each block, was designed to give flexibility in making use of any ODE integrator. Then it is easy to replace the ODE integrator.

Another problem concerning the predictor-corrector methods was the evaluation of adsorption, since the adsorption was calculated explicitly in the simulator. The procedure should be changed to evaluate adsorption implicitly.

A stability analysis was done for the oil bank, and the surfactant front. The former imposes a rather constant limitation on the time step size continuously until the plateau of the oil bank is completely produced, which coincides with surfactant breakthrough. The latter yields unconditional instability continually but only locally. This conclusion is based upon an approximate analysis.

An eigenvalue analysis for the oil bank blocks suggests that the stability is governed by the slope of the water-oil fractional flow curve at the oil bank saturation. This analysis seems to give reasonable criteria for both the fully discrete solution and the semi-discrete solution. The fully discrete solution oscillates to a larger degree when the stability requirement is not satisfied. The semi-discrete method selects a time step size around the limit of stability while the oil bank is being produced, then the time step size is increased thereafter.

An approximate von Neumann stability analysis for the surfactant front blocks showed an unconditional local instability occurred occasionally, causing an oscillation in the history even with a time step size as small as 0.001 PV. This instability is caused by the change in the fractional flow curve due to the reduction of interfacial tension because of the surfactant. It is impossible to

eliminate this instability completely, as long as the continuity equations are solved explicitly, because the change in the fractional flow curve is an essential part of micellar/polymer flooding. However, it may be possible to decrease both the degree of instability and its frequency by changing the equation that gives the residual saturation of the non-wetting phase (oil) as a function of the capillary number.

When the phase behavior environment is Type III, numerical stability may be more difficult to attain. When total composition is within the three phase region, extremely low interfacial tension occurs, which causes a very large change in the fractional flow. Although the stability was analyzed only for two phase flow, the same result may apply. Furthermore, in the three phase region, the microemulsion phase may travel much faster than the other two phases, possibly causing worse instability. It is rather ironical that low interfacial tension contributes to both high oil recovery and to numerical instability during simulation.

NOMENCLATURE

a_i	adsorption parameters
A	binodal curve parameter
A_{p1}, A_{p2}, A_{p3}	polymer viscosity parameter
b_i	adsorption parameters
b_p	permeability reduction parameter
B	binodal curve parameter
C_i	total concentration of component i in mobile phases
C_{ij}	concentration of component i in phase j
\bar{C}_i	volume of adsorbed component i per unit pore volume
\tilde{C}_i	overall concentration of component i in mobile and rock phase
C_{SE}	effective salinity
C_{SEL}	lowest effective salinity for Type III phase behavior
C_{SEU}	highest effective salinity for Type III phase behavior
C_{SEN}	normalized salinity
D_{ij}	effective binary diffusion coefficient of component i in phase j
e_j	relative permeability exponent for phase j
e_{jc}	relative permeability exponent for phase j under the condition of infinite capillary number
e_{jw}	relative permeability exponent for oil/water system
E	distribution curve parameter
F	distribution curve parameter

f_j	fractional flow of phase j
G_{ij}	interfacial tension parameters
J	Jacobian matrix
k	absolute permeability
k_{rj}	relative permeability to phase j
k_{rj}^0	endpoint relative permeability to phase j
k_{rjc}^0	endpoint relative permeability to phase j under the condition of infinite capillary number
k_{rjw}^0	endpoint relative permeability to phase j in oil- water (no-surfactant) system
K_{ij}	dispersion coefficient of component i in phase j
K_ℓ	longitudinal dispersion coefficient
L	length of system
M	total number of phases
NCOMP	total number of components
N_i	mass flux of component i
P	pressure
q	volumetric flow rate
R_k	permeability reduction factor
$R_{k_{\max}}$	maximum value of R_k
S_p	salinity dependence parameter for polymer solution viscosity
S_j	saturation (volume fraction) of phase j
S_{jr}	residual saturation of phase j
S_{jrw}	residual saturation of phase j in water/oil (no

	surfactant) system
t	time
t_D	dimensionless time
T_{jk}	desaturation parameters
u_j	superficial velocity of phase j
w_i	overall mass concentration of component i
x	distance
x_D	dimensionless distance
y	dependent variable of ordinary differential equation

Greek symbols

α_{lj}	longitudinal dispersivity
α_{Dj}	dimensionless dispersivity
α_i	microemulsion viscosity parameter
β	effective salinity parameter
γ	interfacial tension
Δ	difference in operator
ρ_j	density of phase j
ρ_i^o	density of pure component i
ϕ	porosity
τ	permeable media tortuosity factor
λ_{rT}	total relative mobility
μ_j	viscosity of phase j
ω_{ij}	mass fraction of component i in phase j

Subscripts

i	component number
	1 = water
	2 = oil
	3 = surfactant
	4 = polymer
	5 = total anions
	6 = calcium ion
	7 = alcohol
j	phase number
	1 = aqueous (water-rich)
	2 = oleic (oil-rich)
	3 = microemulsion (surfactant-rich)
o	oil
p	polymer or plait point
w	water
mo	microemulsion/oil interface
wm	water/microemulsion interface

APPENDIX A

PHASE BEHAVIOR CONCEPT

It is quite essential to understand phase behavior when one tries to understand the micellar flooding process. Equilibrium ternary diagrams with coordinates surfactant-cosurfactant, brine and oil are commonly used to represent phase behavior. Figure A.1 illustrates three types of generalized phase behavior, called Type II(-), Type III, and Type II(+) following Nelson and Pope². Other authors¹ designate them in different ways. Each diagram has a multiphase region at the bottom separated from a single phase region by the binodal curve. When the total composition is below the binodal curve, more than one phase exists in equilibrium and the saturation of each phase is given as in Figure A.1.

Among the variables that affect the type of diagram observed are effective salinity (including the effect of calcium and other electrolytes), oil composition, surfactant molecular structure, alcohol cosolvent type, and temperature. Any change in those variables which favors the solubility of surfactant in the oil relative to the brine causes the phase environment type to shift in the II(-) to II(+) direction.

In a typical operation of micellar flooding, all those variables which affect the type of diagram are fixed, or assumed to be fixed except electrolytes. Effective salinity may differ among reservoir brine, preflush, surfactant slug, mobility (polymer) buffer and drive

water. Consequently the effect of salinity must be well understood and taken into account in micellar flooding simulators. Also, cation exchange can have a large effect on the electrolytes.

Figure A.2 illustrates the effect of salinity on the phase diagram. As salinity increases, the phase environment type changes from Type II(-) to Type III to Type II(+).

In both Figures A.1 and A.2, phase diagrams are rather simplified and idealized. In real systems, the shape of binodal curve is usually skewed and invariant point may not be a single point. We assume, however, that the idealized phase behavior is a good approximation and employed it in the simulator.

In Type II(-) phase behavior environment, there exists a two phase region wherein microemulsion along the binodal curve is in equilibrium with oil that contains molecularly dispersed surfactant (excess oil). The tie lines which connect two equilibrated phase compositions are of negative slope. The plait point is located on the binodal curve near or on the apex of 100% oil. When the microemulsion composition is at the plait point, the composition of the other phase in equilibrium is also at the plait point and there is no distinction between the phases: there is only one phase with no interface.

In Type II(+) phase behavior environment, there also exists a two phase region, but microemulsion is in equilibrium with excess water rather than excess oil. Consequently, the tie lines are of positive slope and the plait point is located near the apex of 100%

brine.

When the phase behavior environment is Type III, there are three multiphase regions, namely the Type II(+) node, Type II(-) node, and three phase region. The first two are two phase regions and the phase behavior is essentially the same as Type II(+) and Type II(-) as they are called. When total composition is within the triangle below the two phase regions, three phases appear: microemulsion, excess water, and excess oil. The composition of the microemulsion is represented by the invariant point. This fact means the composition of microemulsion does not depend on total composition if it is in the three phase region, although the saturation of each phase does.

As salinity increases within the Type III phase behavior environment, the invariant point moves continuously from the apex of 100% brine to another apex of 100% oil. Even if the phase behavior environment is called Type III, Type II(-) phase behavior may dominate when salinity is low, and Type II(+) may dominate in higher salinity. Thus it should be noted that the appearance of only two phases does not preclude the phase behavior environment being Type III, because of the Type II(+) and Type II(-) nodes.

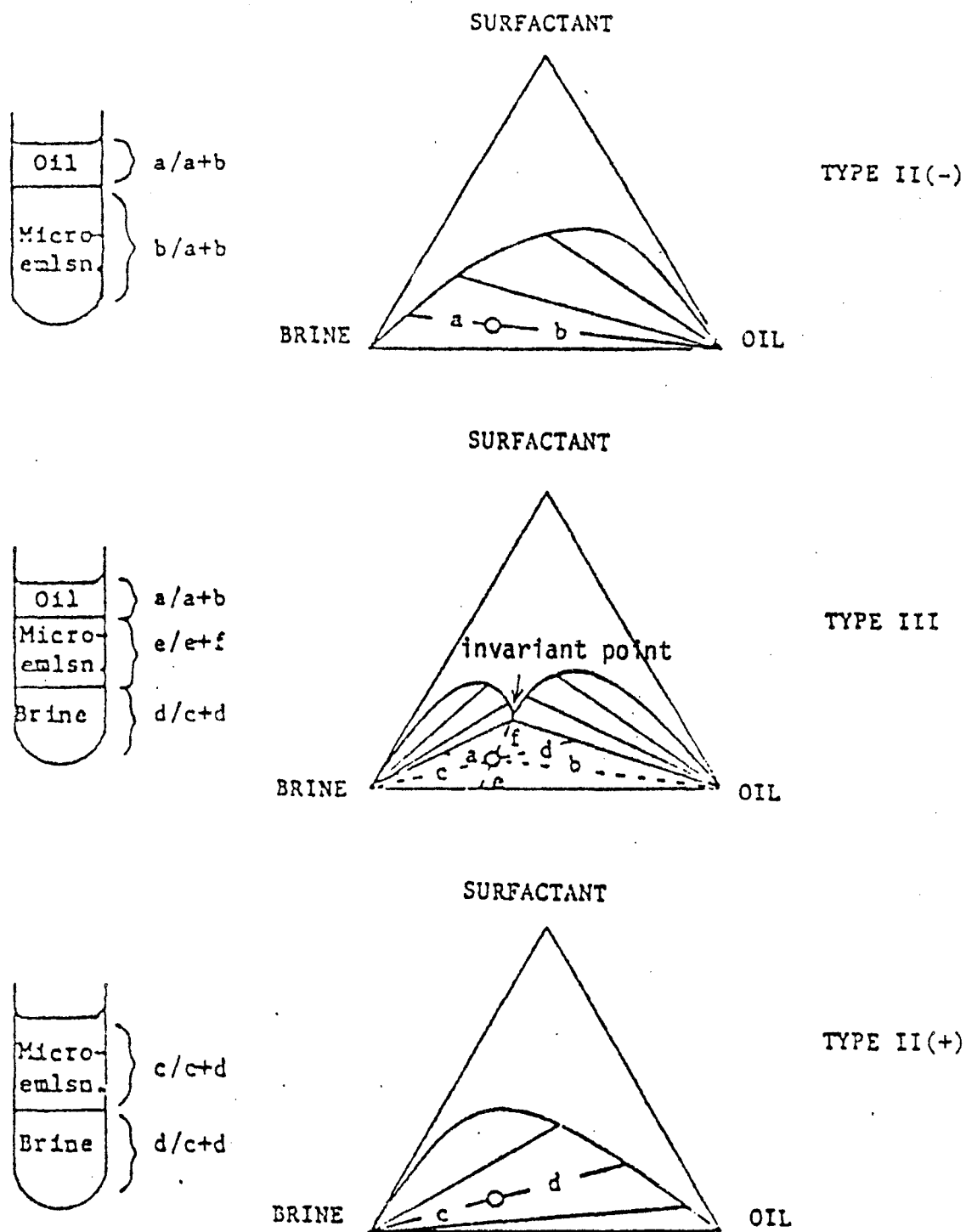


Figure A.1. Ternary representation of diagrams.

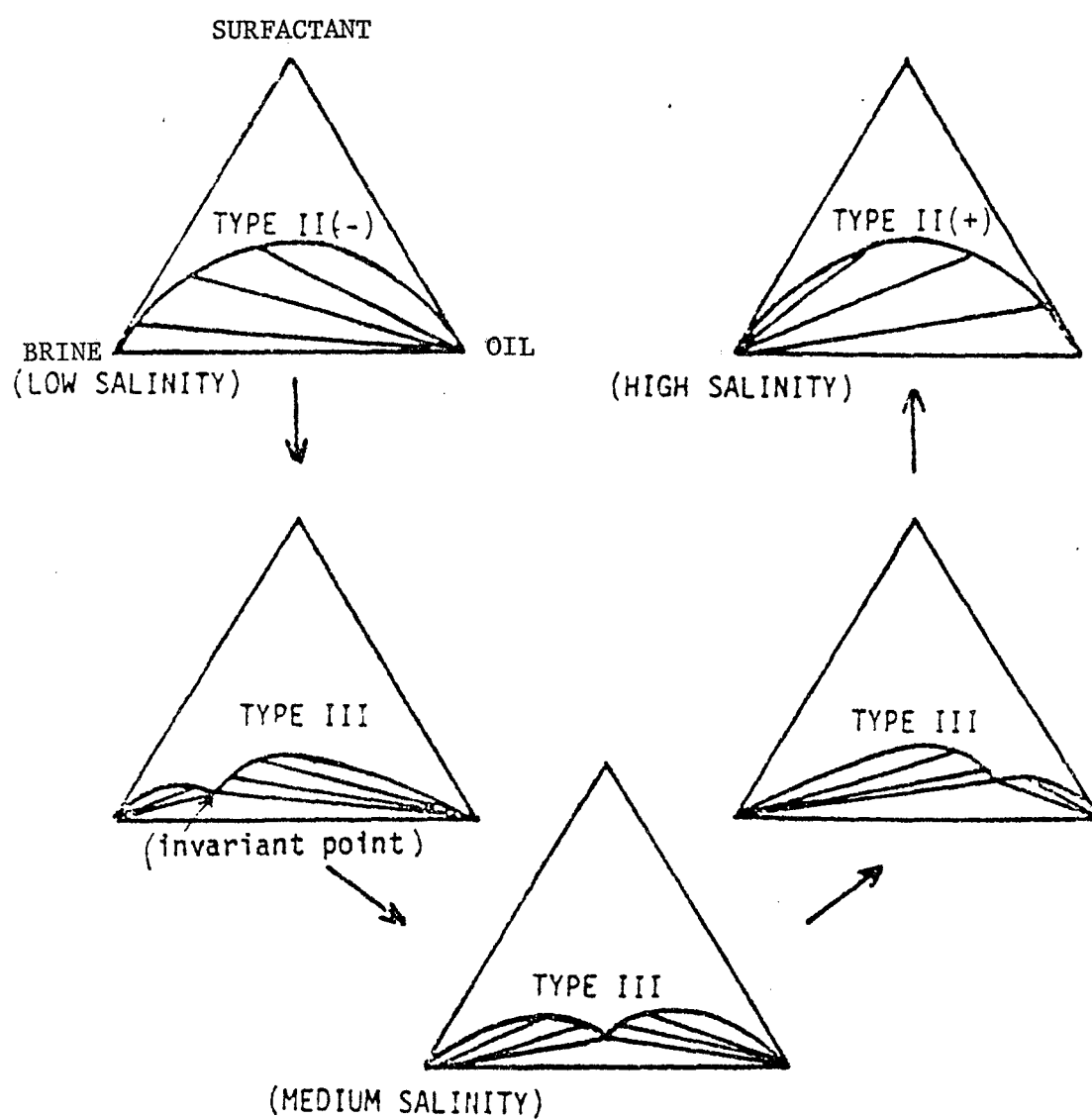


Figure A.2. Effect of salinity on phase diagram

APPENDIX B

DERIVATION OF CONTINUITY EQUATIONS FOR MULTIPHASE MULTICOMPONENT FLOW

Lake et al.⁶³ presented the equations necessary for a complete description of isothermal, multicomponent, multiphase flow in permeable media, which are shown in Table B.1. The first column in Table B.1 gives the differential form of the equation with its name in column two. Column three gives the number of independent scalar equations represented by the equation in column one. Columns four and five give the identity and the number of dependent variables added to the formulation by the equation in column one. In Table B.1, i designates a chemical species ($i=1,\dots,N$), j designates a homogeneous flowing phase ($j=1,\dots,M$), S designates the stationary phase, and D is the number of spatial dimensions ($D \leq 3$).

From assumptions (2) and (10) in Chapter III, the mass conservation equation can be written

$$\frac{\partial W_i}{\partial t} + \frac{\partial N_i}{\partial x} = 0 \quad (\text{B.1})$$

Since in this simulator phase behavior is calculated based on the volume fraction of each component, rather than mass fraction, variables \tilde{C}_i , C_i , and C_{ij} are introduced. Each of them designates the overall volume fraction of component i , the volume fraction in the total fluid, and in phase j , respectively.

$$\tilde{C}_i = (1 - \sum_{i=1}^N \bar{C}_i) C_i + \bar{C}_i \quad (\text{B.2a})$$

$$C_i = \sum_{j=1}^M S_j C_{ij} \quad (\text{B.2b})$$

$$C_{ij} = \rho_j \omega_{ij} / \rho_i^0 \quad (\text{B.3})$$

and

$$\bar{C}_i = \frac{(1 - \phi) \rho_s \omega_{is}}{\phi \rho_i^0} \quad (\text{B.4})$$

\bar{C}_i represents the volume of component i adsorbed per unit pore volume.

If we take into account the volume of adsorbed component, equation (2) in Table B-1 is rewritten

$$W_i = \phi \left(1 - \sum_i^N \bar{C}_i \right) \sum_{j=1}^M \rho_j S_j \omega_{ij} + (1 - \phi) \rho_s \omega_{is} \quad (\text{B-5a})$$

Then combining Eqs. (B.2) through (B.5a)

$$W_i = \phi \rho_i^0 \tilde{C}_i \quad (\text{B.5b})$$

The first term and the second term of Eq. (3) in Table B.1 are usually called the convection term and the dispersion term, respectively.

$$\vec{N}_i = \vec{N}C_i - \vec{N}D_i \quad (\text{B.6a})$$

where

$$\vec{N}C_i = \sum_{j=1}^M \rho_j \omega_{ij} \vec{u}_j \quad (B.7a)$$

$$\vec{N}D_i = \sum_{j=1}^M \phi \rho_j S_j \vec{K}_{ij} \cdot \vec{\nabla} \omega_{ij} \quad (B.8a)$$

Combining Eq. (B.3) and (B.7a)

$$\vec{N}C_i = \rho_i^0 \sum_{j=1}^M c_{ij} \vec{u}_j \quad (B.7b)$$

Since one dimensional flow is being considered, the dispersion coefficient tensor (\vec{K}) is replaced by a scalar $K_{\ell,ij}$, the longitudinal dispersion coefficient, which is usually taken as

$$K_{\ell,ij} = \frac{D_{ij}}{\tau} + \frac{\alpha_{\ell j} |u_j|}{\phi S_j} \quad (B.9a)$$

Since the magnitude of molecular diffusion is much smaller than the dispersion due to convective flow in most flow problems, the first term of Eq. (B.9a) may be neglected. Hence

$$K_{\ell,ij} = \frac{\alpha_{\ell j} |u_j|}{\phi S_j} \quad (B.9b)$$

Substituting Eq. (B.9b) into the 1-D expression of Eq. (B.8a)

$$ND_i = \sum_{j=1}^M \rho_j \alpha_{\ell j} |u_j| \frac{\partial \omega_{ij}}{\partial x} \quad (B.8b)$$

Taking the x positive in flow direction and replacing ω_{ij} by $C_{ij}\rho_i^0/\rho_j$ which is identical to ω_{ij} from Eq. (B.3).

$$\begin{aligned}
 ND_i &= \sum_{j=1}^M \alpha_{lj} \rho_j u_j \frac{\partial}{\partial x} \left(\frac{C_{ij} \rho_i^0}{\rho_j} \right) \\
 &= \rho_i^0 \sum_{j=1}^M \alpha_{lj} \rho_j u_j \frac{\partial}{\partial x} \left(\frac{C_{ij}}{\rho_j} \right) \\
 &= \rho_i^0 \sum_{j=1}^M \alpha_{lj} u_j \left(\frac{\partial C_{ij}}{\partial x} - \frac{C_{ij}}{\rho_j} \frac{\partial \rho_j}{\partial x} \right) \quad (B.8c)
 \end{aligned}$$

Substitution of Eqs. (B.7b) and (B.8c) into Eq. (B.6a) yields

$$N_i = \rho_i^0 u_T \sum_{j=1}^M f_j \left[C_{ij} - \alpha_{lj} \left(\frac{\partial C_{ij}}{\partial x} - \frac{C_{ij}}{\rho_j} \frac{\partial \rho_j}{\partial x} \right) \right] \quad (B.6b)$$

where

$$\begin{aligned}
 u_T &= \sum_{j=1}^M u_j \\
 f_j &= \frac{u_j}{u_T}
 \end{aligned}$$

u_T may change with time but does not change with space. Substituting Eqs. (B.5b) and (B.6b) into Eq. (B.1) and dividing all through by $(\phi \rho_i^0)$

$$\frac{\partial \tilde{C}_i}{\partial t} + \frac{u_T}{\phi} \frac{\partial}{\partial x} \sum_{j=1}^M \left[f_j C_{ij} - \alpha_{lj} f_j \left(\frac{\partial C_{ij}}{\partial x} - \frac{C_{ij}}{\rho_j} \frac{\partial \rho_j}{\partial x} \right) \right] = 0 \quad (B.1b)$$

Introducing dimensionless variables x_D , t_D and a constant α_D , Eq.

(B.1b) is rewritten

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{\partial (f_j C_{ij})}{\partial x_D} - \sum_{j=1}^M \frac{\partial}{\partial x_D} \alpha_{Dj} f_j \left(\frac{\partial C_{ij}}{\partial x_D} - \frac{C_{ij}}{\rho_j} \frac{\partial \rho_j}{\partial x_D} \right) = 0 \quad (\text{B.1c})$$

where ρ_j can be obtained as below

$$\rho_j = \sum_{i=1}^N C_{ij} \rho_i^0 \quad (\text{B.10})$$

If we further assume the differences among pure component densities are insignificant with respect to the dispersion term, and that the dispersion coefficient does not change with space, then

$$\frac{\partial \tilde{C}_i}{\partial t_D} + \sum_{j=1}^M \frac{\partial (f_j C_{ij})}{\partial x_D} - \sum_{j=1}^M \alpha_{Dj} \frac{\partial}{\partial x_D} \left(f_j \frac{\partial C_{ij}}{\partial x_D} \right) = 0 \quad (3.1)$$

Table B.1

Summary of Equations for Isothermal
Fluid Flow in Permeable Media

Equation	Name	Number Independent Scalar Equations	Dependent Variable	
			Identity	Number
(1) $\frac{\partial W_i}{\partial t} + \vec{\nabla} \cdot \vec{N}_i = R_i$	Mass conservation	N	W_i, R_i, \vec{N}_i	2N+ND
(2) $W_i = \phi \sum_{j=1}^M \rho_j S_{ij} \omega_{ij} + (1-\phi) \rho_s \omega_{is}$	Accumulation term	N-1	$\rho_j, S_{ij}, \omega_{ij}, \omega_{is}$	2M+MN+N
(3) $\vec{N}_i = \sum_{j=1}^M (\rho_j \omega_{ij} \vec{u}_j - \phi \rho_j S_{ij} \vec{K}_{ij} \cdot \vec{\nabla} \omega_{ij})$	Flux term	ND	\vec{u}_j	MD
(4) $R_i = \phi \sum_{j=1}^M S_{ij} r_{ij} + (1-\phi) r_{is}$	Source term	N-1	r_{ij}, r_{is}	MN+N
(4a) $\sum_{i=1}^N R_i = 0$	Total reaction definition	1	-	-
(5) $\vec{u}_j = -\lambda_{rj} \vec{\nabla} P_j + \rho_j \vec{g}$	Darcy's Law	MD	λ_{rj}, P_j	2M
(6) $\lambda_{rj} = \lambda_{rj}(S, \omega, \vec{u}_j, \vec{x})$	Relative mobility	M	-	-
(7) $P_j - P_\ell = P_{cjl}(S, \omega, \vec{x})$	Capillary pressure	M-1	-	-
(8) $\sum_{i=1}^N \omega_{ij} = 1$	Mass fraction definition	M	-	-
(8a) $\sum_{i=1}^N \omega_{is} = 1$	Stationary phase mass fraction definition	1	-	-
(9) $\sum_{j=1}^M S_j = 1$	Saturation definition	1	-	-
(10) $r_{ij} = r(\omega, P_j)$	Homogeneous kinetic reaction rates	(N-1)M	-	-
(10a) $r_{is} = r_{is}(\omega_{is})$	Stationary phase reaction rates	N-1	-	-
(10b) $\sum_{i=1}^N r_{ij} = 0$	Total phase reaction	M	-	-

Equation	Name	Number Independent Scalar Equations	Dependent Variable	
			Identity	Number
(10c) $\sum_{i=1}^N r_{is} = 0$	Stationary phase total reaction definition	1	-	-
(11) $\omega_{ij} = \omega_{ij}(\omega_{ik}, k \neq j)$	Equilibrium relations as phase balances	$N(M-1)$	-	-
(11a) $\omega_{is} = \omega_{is}(\omega)$	Stationary phase equi- librium relations as balances	N	-	-
(12) $\rho_j = \rho_j(\omega, P_j)$	Equation of state	M	-	-

Total independent equations = $D(M+N) + 2MN + 4M + 4N$

Total dependent variables = $D(M+N) + 2MN + 4M + 4N$

APPENDIX C
COMPUTER PROGRAM

The micellar/polymer flooding simulator used in this research consists of one main program, twenty five subprograms and one dummy program. The name of each program and its role is as follows:

MAIN	Main program. Drives all subprograms.
INPUT	Read and print out input data, calculate or specify some parameters, and set up initial conditions.
OUTPUT	Depending on the pore volume injected, call some of subprograms HPRINT, PROF, PRFPLOT, HISPLOT, and MATBAL. The boundary condition at injector (injected compositions) is changed as necessary.
HPRINT	Print out production history and store the values necessary to plot history.
PROF	Print out profile.
PRFPLOT	Plot profiles.
HISPLOT	Plot histories.
SOLVE	Solve continuity equations with fully discrete method. Cumulative production and relative pressure drop are also calculated.
SOLVE1	Solve continuity equations with semi-discrete method. Cumulative production and relative pressure drop are also calculated.

FCNJ	Dummy subprogram needed when DGEAR is used.
RK12	ODE integrator with a RKF algorithm.
RK1	ODE integrator with a RK algorithm.
DGEAR	IMSL Library ⁵⁹ . ODE integrator, which allow the use of either Adams' methods or backward differentiation methods.
DER	Calculate the change in concentration as derivatives with respect to time.
PROPRTY	Calculate viscosities, fractional flows and mobilities
RELPERM	Calculate residual saturations and relative permeabilities
POPE	Three phase flow model used in original simulator
HIRA	Three phase flow model presented by G. Hirasaki
LAKE	Three phase flow model presented by L. Lake
PHCOMP	Calculate phase concentrations and saturations according to ternary phase behavior. Interfacial tension is also calculated.
TIELINE	Find a root of non-linear equation with bisection method
TRY	Using the equations of binodal and distribution curves, calculate phase composition with one degree of freedom
IONCNG	Calculate cation exchange
COMPLEX	Calculate cation exchange in case where surfactant complex forms.
CHEMADN	Calculate surfactant adsorption.
POLYADN	Calculate polymer adsorption
MATBAL	Gives final condition and material balance error

As is shown in Figure C.1, there are two loops in main program. Depending on the value of ISOLV specified in input data, either the semi-discrete method or fully-discrete method is selected to solve continuity equations. NSTOP is set to be one in subprogram OUTPUT, when total pore volume desired has been injected.

Figure C.2 and C.3 show which subprogram is called and where, for fully-discrete and semi-discrete solutions respectively. Most of the subprograms which are the essential part of the micellar/polymer flooding simulator are common for both methods. Since subprogram DER was separated from SOLVE, ODE integrators such as DGEAR, RK12, or RK1 can be easily replaced by other methods.

In subprogram DER, special care is taken for the irreversibility of adsorptions. Since predicted time step size may be rejected and all calculations may be repeated with smaller time step size, the adsorption calculated at the last time step are stored and can be used when the rejection occurs. This is done by assigning different values to IADS depending on the situation.

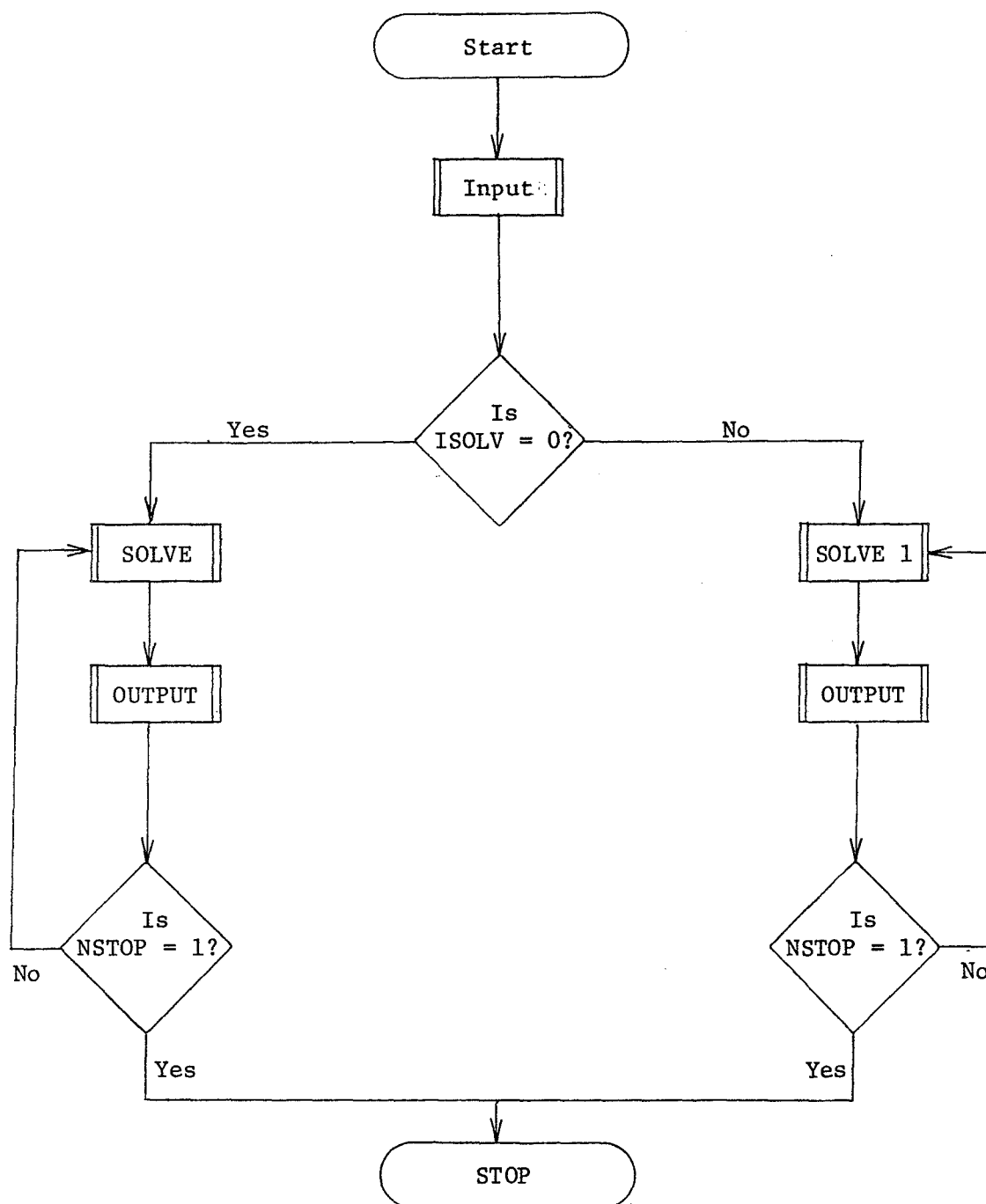


Figure C.1. Flow chart of main program.

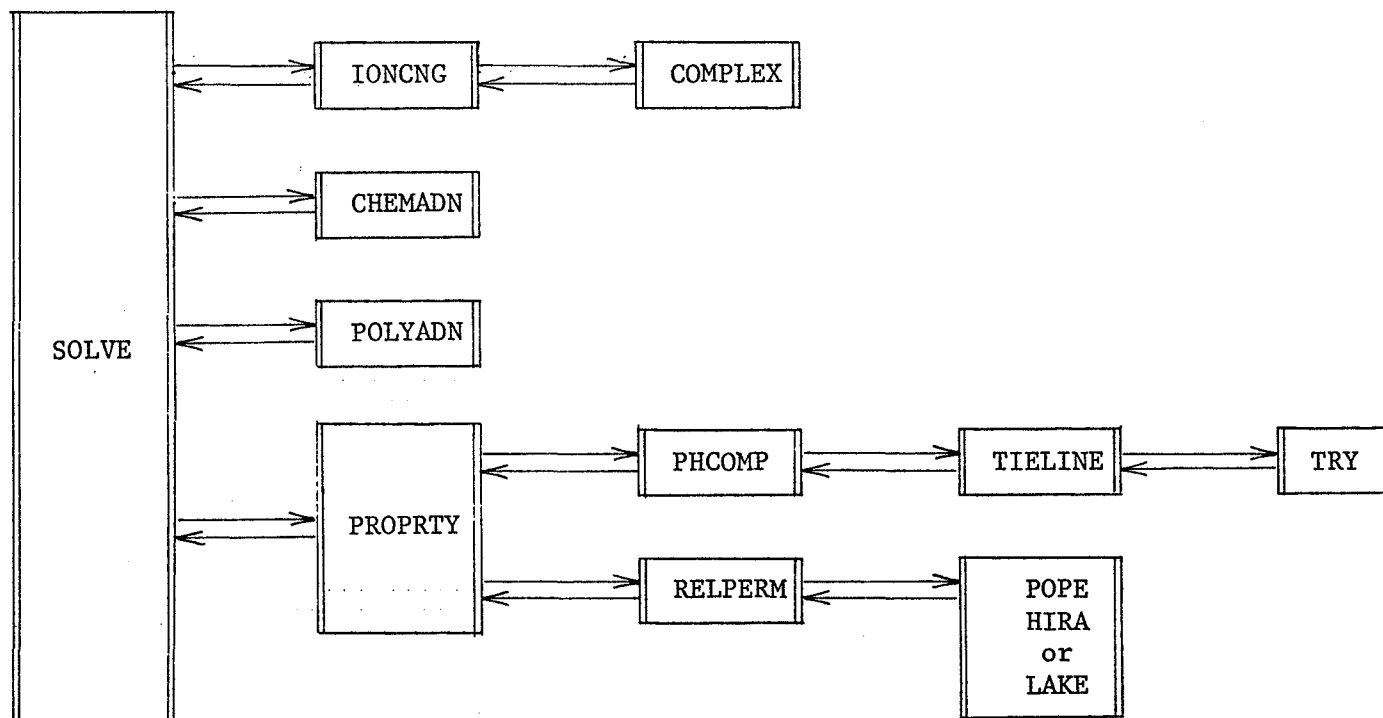


Figure C.2. Subprogram calling sequences in fully-discrete solution.

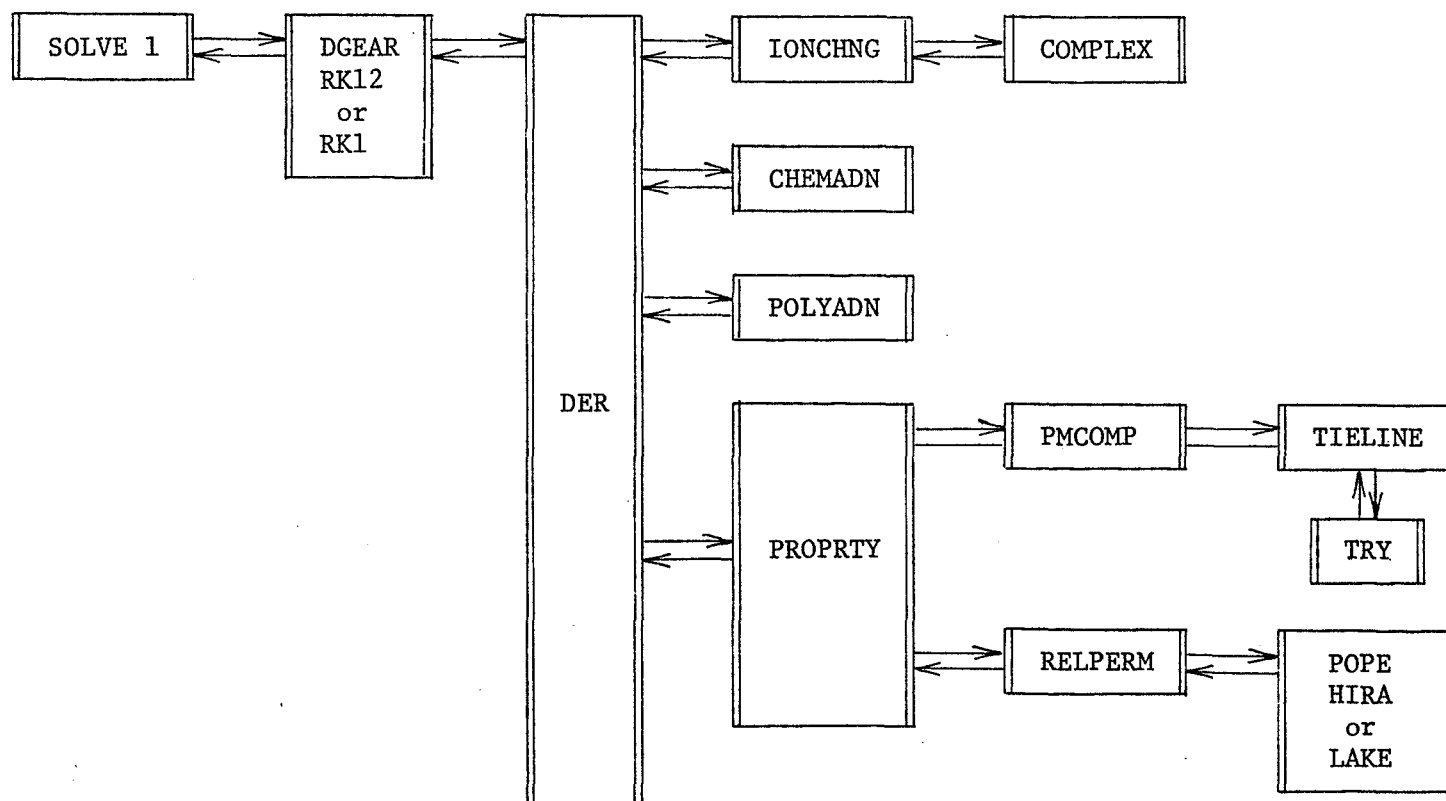


Figure C.3. Subprogram calling sequences in semi-discrete solution.

```

C*****
C THIS IS A ONE-DIMENSIONAL MICELLAR/POLYMER FLOODING SIMULATOR,
C AT MOST THREE (AQUEOUS, OLEIC, AND MICROEMULSION) PHASES ARE
C CONSIDERED.
C MAJOR PHYSICAL FEATURES IN THIS SIMULATOR ARE AS FOLLOWS:
C   PHASE BEHAVIOR
C   INTERFACIAL TENSION
C   TRAPPING FUNCTION
C   VISCOSITY
C   RELATIVE PERMEABILITY
C   ADSORPTION
C   DISPERSION
C   INACCESSIBLE PORE VOLUME
C   ALCOHOL EFFECT
C   CATION EXCHANGE AND SURFACTANT COMPLEX
C*****
C DEFINITION OF COMPONENT AND PHASE NUMBERS
C
C   I = COMPONENT NUMBER
C       1 = WATER
C       2 = OIL
C       3 = SURFACTANT
C       4 = POLYMER
C       5 = TOTAL NONSORBING ANIONS
C       6 = CALCIUM
C       7 = ALCOHOL
C       8 = CALCIUM-SURFACTANT COMPLEX
C   J = PHASE NUMBER
C       1 = AQUEOUS(WATER-RICH)
C       2 = OLEIC(OIL-RICH)
C       3 = MICROEMULSION(MOST SURFACTANT RICH PHASE)
C*****
C EXPLANATION OF INPUT DATA
C
C   AD31,AD32 = SURFACTANT ADSORPTION PARAMETERS
C   AD32      = SALINITY DEPENDENCE SURFACTANT ADSORPTION PARAMETER
C   A40,A40D = POLYMER ADSORPTION PARAMETERS
C   ABPERH   = ABSOLUTE PERMEABILITY (DARCY)
C   ALPHA1 - ALPHAS = PHASE VISCOSITY PARAMETERS
C   AP1 - AP3 = POLYMER-RICH PHASE VISCOSITY PARAMETERS
C   BRK      = PERMEABILITY REDUCTION PARAMETER
C   CIN(I,J) = CONCENTRATION OF COMPONENT I IN J-TH INJECTED SLUG
C   CSEL     = LOWEST EFFECTIVE SALINITY FOR TYPE III PHASE BEHAVIOR
C   CSEU     = HIGHEST EFFECTIVE SALINITY FOR TYPE III PHASE BEHAVIOR
C   CSEI     = CUT OFF SALINITY FOR THE SALINITY EFFECT ON POLYMER
C   C2PLC    = OIL CONCENTRATION AT PLAIT POINT FOR TYPE II(+)
C   C2PRC    = OIL CONCENTRATION AT PLAIT POINT FOR TYPE II(-)
C   C3MAXN,C3MAX1,C3MAX2 = MAXIMUM HEIGHT OF BINODAL CURVE AT THE
C                           NORMALIZED SALINITY OF 0, 1, AND 2
C   CS11,C611 = INITIAL CONCENTRATIONS OF ANIONS AND CALCIUM IN
C                           AQUEOUS PHASE
C   DISP     = DIMENSIONLESS LONGITUDINAL DISPERSIVITY
C   DTMAX    = MAXIMUM TIME STEP SIZE ALLOWED IN SEMI-DISCRETE SOLUTION
C   ERR      = RELATIVE ERROR TOLERANCE FOR SEMI-DISCRETE METHODS
C   E1,E2    = RELATIVE PERMEABILITY EXPONENT FOR WATER AND OIL
C                           IN WATER-OIL SYSTEM
C   EPH13,EPH14 = EFFECTIVE POROSITY FOR SURFACTANT AND POLYMER

```

```

C   FFDV     = THE RATIO OF TIME STEP SIZE OVER SPATIAL GRID SIZE
C   GAMHF    = SHEAR RATE AT WHICH VISCOSITY EQUALS ONE HALF OF
C                           NO SHEAR VISCOSITY
C   G11 - G13 = INTERFACIAL TENSION PARAMETERS FOR TYPE II(+)
C   G21 - G23 = INTERFACIAL TENSION PARAMETERS FOR TYPE II(-)
C   ICT       = NUMBER OF GRID BLOCKS
C   ICTL,ICTU = PRESSURE IS MEASURED BETWEEN ICTL-TH AND ICTU-TH BLOCK
C   ID        = SITE I.D. FOR THE RELEASE OF PLOTS
C   IPERM     = PARAMETER TO SPECIFY THREE PHASE FLOW MODEL
C   ISEM      = PARAMETER TO SPECIFY O.D.E. INTEGRATOR
C   ISOLV     = PARAMETER TO SPECIFY THE NUMERICAL SOLUTION TECHNIQUE
C   NCOMP     = NUMBER OF COMPONENTS TO BE CONSIDERED
C   NSLUG     = NUMBER OF SLUGS INJECTED
C   METH      = PARAMETER TO SPECIFY PREDICTOR-CORRECTOR METHOD
C   MITER     = PARAMETER TO SPECIFY THE ITERATION SCHEME FOR A
C                           PREDICTOR-CORRECTOR METHOD
C   PHI       = POROSITY
C   POWN      = SHEAR RATE DEPENDENCE PARAMETER FOR POLYMER VISCOSITY
C   PIRN,P2RW = ENDPOINT RELATIVE PERMEABILITY TO WATER AND OIL IN
C                           WATER/OIL SYSTEM
C   PIRC,P2RC = END POINT RELATIVE PERMEABILITY TO AQUEOUS AND OLEIC
C                           PHASE UNDER THE CONDITION OF INFINITE CAPILLARY NUMBER
C   QV        = CATION EXCHANGE CAPACITY (MEQ/ML PORE VOLUME)
C   RCSE      = PARAMETER FOR THE CALCULATION OF EFFECTIVE SALINITY
C   RKMAX     = MAXIMUM VALUE OF PERMEABILITY REDUCTION FACTOR
C   SSLOPE    = SALINITY DEPENDENCE PARAMETER FOR POLYMER VISCOSITY
C   S1,S2     = INITIAL SATURATION OF WATER AND OIL
C   S1RW,S2RW = RESIDUAL SATURATION OF WATER AND OIL IN WATER/OIL
C                           SYSTEM
C   TITLE     = INFORMATION FOR YOURSELF (DOES NOT AFFECT COMPUTATION)
C   T11,T12   = DESATURATION PARAMETERS FOR AQUEOUS PHASE
C   T21,T22   = DESATURATION PARAMETERS FOR OIL PHASE
C   UT        = SUPERFICIAL (DARCY) VELOCITY OF TOTAL PHASE
C                           (USED ONLY FOR SHEAR RATE EFFECT ON POLYMER)
C   VIN(N)    = CUMULATIVE AMOUNT OF INJECTED FLUID AFTER THE COMPLETION
C                           OF N-TH SLUG INJECTION
C   VIS1,VIS2 = VISCOSITY OF WATER AND OIL
C   VT        = TOTAL AMOUNT OF INJECTED FLUID (PV)
C   XEND      = SHOULD BE SMALL NUMBER SUCH AS 0.0001 (USED IN DGEAR)
C   XIFTH     = LOGARITHM OF INTERFACIAL TENSION BETWEEN WATER AND OIL
C   XK96,XK86,XKAT = MASS ACTION EXCHANGE CONSTANTS
C   XKC       = EQUILIBRIUM CONSTANT USED IN THE CALCULATION OF COMPLEX
C   YBIAS     = CUT OFF VALUE FOR ESTIMATED ERROR TO AVOID EXCESSIVELY
C                           SMALL TIME STEP (USED IN RK12 AND RK1)
C*****
C EXPLANATION OF OPTION PARAMETERS
C
C   NUMERICAL SOLUTION TECHNIQUE
C   ISOLV = 0 FOR FULLY-DISCRETE FORWARD EULER METHOD
C           1 FOR SEMI-DISCRETE METHOD
C
C   ORDINARY DIFFERENTIAL EQUATION INTEGRATOR
C   (REQUIRED IF ISOLV=1)
C   ISEM = 1 FOR RK12 (RUNGE-KUTTA-FEHLBERG ALGORITHM)
C         2 FOR RK1 (RUNGE-KUTTA ALGORITHM)
C         3 FOR DGEAR (ADAMS OR GEAR'S METHOD)
C
C   PREDICTOR-CORRECTOR METHOD
C   (REQUIRED IF ISOLV=1 AND ISEM=3)

```

```

C      METH = 1 FOR ADAMS# METHODS
C          = 2 FOR GEAR#S METHODS
C
C      ITERATION SCHEME FOR PREDICTOR-CORRECTOR METHOD
C      (REQUIRED IF ISOLV=1 AND ISEM=3)
C      MITER = 0 FOR FUNCTIONAL(FIXED POINT) ITERATION
C          = 1 FOR A CHORD METHOD WITH THE JACOBIAN SUPPLIED BY
C              THE USERS
C          = 2 FOR A CHORD METHOD WITH THE JACOBIAN CALCULATED
C              INTERNALLY BY FINITE DIFFERENCES
C          = 3 FOR A CHORD METHOD WITH THE JACOBIAN REPLACED BY
C              A DIAGONAL APPROXIMATION BASED ON A DIRECTIONAL
C              DERIVATIVE
C
C      THREE PHASE FLOW MODEL
C      IPERM = 1 FOR POPE#S MODEL
C          = 1 FOR HIRASAKI#S MODEL
C          = 2 FOR MODIFIED HIRASAKI#S MODEL (NOT COMPLETED YET)
C          = 3 FOR LAKE#S MODEL
C
C      FOR SITE I.D. TO SPECIFY WHERE THE PLOTS SHOULD BE RELEASED,
C      CONSULT TAUNHS USER#S REFERENCE
C
C*****
C      EXPLANATION OF VARIABLES IN COMMON BLOCKS OR FORMAL PARAMETERS
C      SUBSCRIPTS ARE USED AS FOLLOWS :
C          I = COMPONENT NUMBER
C          J = PHASE NUMBER
C          K = BLOCK NUMBER
C
C      A11,A12 = MINODAL CURVE PARAMETERS USED WHEN CSE,GT,CSEOP
C      A21,A22 = MINODAL CURVE PARAMETERS USED WHEN CSE,LE,CSEOP
C      A30 = SURFACTANT ADSORPTION PARAMETER (=AD31+AD32+CSE)
C      A3DS(K) = OLD VALUE OF ASD NEEDED FOR IRREVERSIBILITY
C      R1,R2,R3 = MINODAL CURVE PARAMETERS (FIXED TO BE MINUS UNITY
C              INSIDE THE SIMULATOR)
C      C(I,J,K) = CONCENTRATION OF COMPONENT I IN PHASE J AT K-TH BLOCK
C      C(I,4,K) = TOTAL CONCENTRATION OF COMPONENT I IN MOBILE PHASE OR
C              OVERALL CONCENTRATION OF COMPONENT I IN BOTH MOBILE
C              AND ROCK PHASE
C      CSE(K) = EFFECTIVE SALINITY
C      CSEOP = OPTIMAL SALINITY
C      C3PH = SURFACTANT CONCENTRATION IN MOST SURFACTANT RICH PHASE
C      C4PH = POLYMER CONCENTRATION IN MOST POLYMER RICH PHASE
C      C3ADSS(K) = VOLUME OF SURFACTANT ADSORBED PER UNIT PORE VOLUME
C      C4ADSS(K) = CONCENTRATION OF POLYMER IN ROCK PHASE
C      C6ADSS(K) = CONCENTRATION OF CALCIUM IN ROCK PHASE
C      C8ADSS(K) = CONCENTRATION OF COMPLEX IN ROCK PHASE
C      C6MATS(K) = CONCENTRATION OF COUNTER ION OF ADSORBED SURFACTANT
C      C5MAX,C6MAX = MAXIMUM CONCENTRATION OF ANION AND CALCIUM IN THE
C              PAST (USED TO NORMALIZE CONCENTRATION FOR PLOT)
C      DC3 = SMALL INCREMENT USED IN TOUGH
C      EN = CUMULATIVE OIL PRODUCTION
C      OVP = TIME STEP SIZE (TV)
C      DISP(J) = DIMENSIONLESS LONGITUDINAL DISPERSIVITY
C      DTOLD = TIME STEP SIZE AT THE LAST STEP (FOR SEMI-DISCRETE)
C      FF(J,K) = FRACTIONAL FLOW
C      F1 - F3 = DISTRIBUTION CURVE PARAMETER (FIXED TO BE UNITY
C              INSIDE THE SIMULATOR)
C      IADS = SWITCH USED TO DEAL WITH IRREVERSIBLE SURFACTANT
C              ADSORPTION WHEN RK1 OR RK12 IS USED

```

```

C      ICT1 = ICT+1
C      ICT2 = ICT+2
C      IEVA = NUMBER OF FUNCTION EVALUATION WITH SFMI-DISCRETE METHOD
C      IM = COUNTER USED TO PUT TITLES IN HISTORY PRINT
C      IPASS = SWITCH USED TO CONVERT C(I,4,K) TO 1-D ARRAY IN SOLVE1
C      IPV = CUMULATIVE NUMBER OF TIME STEP
C      ISHEAR = INDICATOR FOR SHEAR RATE EFFECT ON POLYMER
C      K,KK = BLOCK NUMBER
C      NEH = NUMBER OF EQUATIONS (NCOMP+ICT)
C      NHEJ = NUMBER OF PREDICTED TIME STEP SIZE REJECTION
C      NPHASE(K) = NUMBER OF PHASES AT THE BLOCK
C              (SET TO BE 4 FOR THE LEFT NODE OF TYPE III)
C              (SET TO BE 5 FOR THE RIGHT NODE OF TYPE III)
C      NSTOP = INDICATOR WHETHER THE JOB IS COMPLETED
C      P(I) = CUMULATIVE PRODUCTION
C      PERM(J,K) = RELATIVE PERMEABILITY
C      PHT(K) = TOTAL RELATIVE MOBILITY
C      PHTLU = TOTAL RELATIVE MOBILITY AT INITIAL CONDITION
C      PRES(K) = NORMALIZED PRESSURE DROP (=PHTLU/PHT(K))
C      PREMAX = MAXIMUM PRESSURE DROP
C      PRESUM = TOTAL PRESSURE DROP (SUM OF PRES(K))
C      PS = CUMULATIVE PRODUCTION OF SURFACTANT COMPLEX
C      RTEMAX = MAXIMUM LOCAL TRUNCATION ERROR (RELATIVE) ESTIMATED
C      S(J,K) = SATURATION
C      SN(J,K) = SATURATION IN REDUCED SATURATION SPACE
C      SRED(J,K) = RESIDUAL SATURATION
C      SJR = SJR = RESIDUAL SATURATION (IDENTICAL TO SRED(J,K))
C      T31 - T32 = DESATURATION PARAMETER FOR MICROEMULSION
C              (USED ONLY IN POPE#S RELATIVE PERMEABILITY MODEL)
C              (SET TO ZERO INSIDE THE SIMULATOR)
C      VP = CURRENT TIME (CUMULATIVE INJECTION)
C      VPI = TIME WHEN THE INJECTED SLUG IS CHANGED NEXT
C      VIS = VISCOSITY
C      XICT = GRID SIZE (INVERSE OF ICT)
C      XIFT1(K) = LOG OF IFT BETWEEN AQUEOUS AND MICROEMULSION PHASE
C      XIFT2(K) = LOG OF IFT BETWEEN MICROEMULSION AND OLEIC PHASE
C      XIFT3(K) = MIN(XIFT1,XIFT2)
C      ZE(I) = TOTAL AMOUNT INITIALLY EXISTED
C      ZI(I) = TOTAL AMOUNT INJECTED
C
C*****

```

```

      PROGRAM MAIN(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,PLOTR)
C-----
C THIS MAIN PROGRAM DRIVES ALL SUBPROGRAMS.
C DEPENDING ON ISOLV, CONTINUITY EQUATIONS ARE SOLVED EITHER WITH
C FULLY-DISCRETE OR SEMI-DISCRETE METHOD.
C   ISOLV = 0 FOR FULLY-DISCRETE METHOD
C   ISOLV = 1 FOR SEMI-DISCRETE METHOD
C-----
C
      COMMON/MAIN/ISOLV,NSTOP,TD
C
      CALL INPUT
      IF (ISOLV.EQ.0) GO TO 10
      CALL SOLVE1
      CALL OUTPUT
      IF (NSTOP.NE.1) GO TO 20
      GO TO 30
      CALL SOLVE
      CALL OUTPUT
      IF (NSTOP.NE.1) GO TO 10
      CONTINUE
      STOP
      END

```

```

      SUBROUTINE INPUT
C-----
C THIS SUBPROGRAM READS AND PRINT INPUT DATA.
C SOME PARAMETERS ARE CALCULATED IN THIS SUBPROGRAM.
C INITIAL CONDITIONS ARE ALSO SET IN THIS SUBPROGRAM.
C-----
      DIMENSION TITLE(20)
C
      COMMON/MAIN/ISOLV,NSTOP,TD
      COMMON/NO/ICT,ICT1,ICT2,XICT,NCOMP
      COMMON/SYSTEM/UT,ABPERM,PHT,EPH13,EPH14,DISPJ(4)
      COMMON/SEMI1/DTHAX,ERR,YBIAS,IPASS
      COMMON/SEMI3/HEQ,ISEM,XEND,METH,MITER
      COMMON/IN/VIN(10),CIN(7,10)
      COMMON/SOL/C(7,4,02),S(3,02),FF(3,02),NPHASE(02)
      COMMON/CSE/CSE(02),CSEL,CSEU,RCSE,CSEOP
      COMMON/PHASE/F1,F2,F3,R1,B2,R3,C2PLC,C2PRC
      COMMON/A/A11,A12,A21,A22
      COMMON/PRODIN/ER,P(7),P1,ZI(7),ZE(7),S2
      COMMON/ADSSR/C3ADSS(02),C4ADSS(00),C6ADSS(00),C6HATS(00)
      COMMON/C3/C3ADSS(04),CC0(02)
      COMMON/PERM/IPER11,P1RW,P2RW,E1,E2,P1RC,P2RC
      COMMON/IFT/G11,G12,G13,G21,G22,G23
      COMMON/XIFT/XIFT1(02),XIFT2(02),XIFT3(02),XIFTW
      COMMON/TPAP/T11,T12,T21,T22,T31,T32,S1PW,S2RW,PHT(00)
      COMMON/ALPHA/ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5
      COMMON/CSEVIS/VIS1,VIS2,AP1,AP2,AP3,SSLOPE
      COMMON/SHEVIS/GAMHF,POWH,CSE1,RKMAX,RRK,ISHEAR
      COMMON/CHEMAD/C3PH,A3D,B3D
      COMMON/POLYAD/C4PH,A4D,B4D
      COMMON/PRESS/PHTLU,PRESUM,ICTL,ICTU,PRES(02)
      COMMON/HMR/XKC,XK96,XK98,XKHAT,OV
      COMMON/ION/FFDV,DC3,K
      COMMON/A3D/A3DS(00),AD31,AD32
      COMMON/INJECT/DVP,VP,VT,VPI
      COMMON/CPLDT/C5MAX,C6MAX
C
C***** READ INPUT DATA *****
      READ 100,(TITLE(I),I=1,20)
      READ 112,NSLUG
      READ 112,ISOLV,ISEM,METH,MITER,IPERM,TD
      READ 111,DTHAX,ERR,YBIAS,XFND
C INITIAL CONDITIONS AND SYSTEM VALUES
      READ 110,VT,FFDV,NCOMP,ICT,ICT1,ICTU
      READ 111,UT,ABPERM,PHT,EPH13,EPH14,DISPJ
      READ 111,C3IT,C4I1,S1,S2,S1RW,S2RW
C
C IFT AND TRAPPING FUNCTION PARAMETERS
      READ 111,G11,G12,G13,G21,G22,G23
      READ 111,T11,T12,T21,T22,XIFTW
C
C PHASE VISCOSITY PARAMETERS
      READ 111,ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5
      READ 111,VIS1,VIS2,AP1,AP2,AP3,SSLOPE
      READ 111,GAMHF,POWH,CSE1,RKMAX,RRK
C
C RELATIVE PERMEABILITY PARAMETERS
      READ 111,P1RW,P2RW,E1,F2,P1RC,P2RC
C
C PHASE BEHAVIOR PARAMETERS
      READ 111,C3MAX,C3MAX1,C3MAX2

```



```

      READ 111,C2PLC,C2PPC,CSEL,CSEU
C
C ADSORPTION AND IONEXCHANGE PARAMETERS
      READ 111,OV,RCSE,AD31,AD32,R3D
      READ 111,A4D,B4D,XK94,XK86,XKC,XKHAT
C
C INJECTED SLUG COMPOSITION
      DO 10 I=1,NSLUG
      10 READ 111,VIN(N),(CIN(I,N),I=1,NCOMP)
C
C***** CALCULATE PARAMETERS BASED ON INPUT DATA *****
      F1=F2=F3=1.0
      T1=T2=T3=0.0
      H1=H2=H3=-1.0
      DC3=1.0
      AA0=((2.0*C3MAXP)/(1.0-C3MAXP))*2
      AA1=((2.0*C3MAX1)/(1.0-C3MAX1))*2
      AA2=((2.0*C3MAX2)/(1.0-C3MAX2))*2
      CSEOP=0.5*(CSEU+CSEL)
      A21=AA0
      A22=((AA1-AA0)/CSEOP)
      A11=7.0*AA1-AA2
      A12=((AA2-AA1)/CSEOP)
C
      DVP=FFDV/FLOAT(ICT)
      ICT1=ICT+1
      ICT2=ICT+2
      XICT=1./FLOAT(ICT)
      IF(UT.LE.1.0E-8)ISHEAR=1
      IF(ABS(POWH-1.0).LE.0.01)ISHEAR=1
      NEQ=ICT*NCOMP
C
      C5MAX=C511
      C6MAX=C611
      DO 12 N=1,NSLUG
      IF(CIN(5,N).GT.C5MAX)C5MAX=CIN(5,N)
      IF(CIN(6,N).GT.C6MAX)C6MAX=CIN(6,N)
      12 CONTINUE
C
C INITIAL PERMEABILITIES AND FRACTIONAL FLOW
      SR=(S1-S1R)/(1.0-S1PW-S2RW)
      IF(SR.LT.0.0)SR=0.0
      IF(SR.GT.1.0)SR=1.0
      PERM1=PIRW*SR**E1
      PERM2=P2RW*(1.4-SR)**E2
      PHTLU=PERM1/VIS1+PERM2/VIS2
      FF1=PERM1/VIS1/PHTLU
      FF2=PERM2/VIS2/PHTLU
C
C INITIAL AMOUNT OF CALCIUM ADSORBED
      C6IAD=0.0
      IF(CA11.LE.0.0)GO TO 54
      C9=C511-CA11
      H96=XK96*C9**2/C611
      IF(OV.LT.0.00001)GO TO 52
      C6IAD=5.0*(2.0*OV+R96-SQRT(4.0*OV+R96**2))
      52 RHAT=XKHAT*C9**2/C611
      54 CONTINUE
C
C***** PRINT INPUT DATA *****
      PRINT 200,(TITLE(I),I=1,74)
      IF(ISOLV.NE.1)GO TO 14

```

```

      PRINT 201
      IF(ISEM.EQ.1)PRINT 202
      IF(ISEM.EQ.2)PRINT 203
      IF(ISEM.EQ.3)PRINT 205
10 CONTINUE
      IF(IPERM.EQ.2)PRINT 240
      IF(IPERM.EQ.1)PRINT 241
      IF(IPERM.EQ.2)PRINT 242
      IF(IPERM.EQ.3)PRINT 243
      PRINT 214
      PRINT 250,NSLUG
      PRINT 251,ISOLV,ISEM,METH,MITER,IPERM,IO
      PRINT 211,V1,FFDV,NCOMP,ICT,ICTL,ICTU
      PRINT 212,UT,ABPERM,PHT,EPH13,EPH14,DISP
      PRINT 213,C511,C611,S1,S2,S1RW,S2RW
      PRINT 214,G11,G12,G13,G21,G22,G23
      PRINT 215,T11,T12,T21,T22,XIFTW
      PRINT 216,ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5
      PRINT 217,VIS1,VIS2,AP1,AP2,AP3,ASLOPE
      PRINT 218,G4HMF,POHNF,CSE1,RKMAX,BRK
      PRINT 219,PIRW,P2RW,E1,E2,PIRC,P2RC
      PRINT 221,C3MAXP,C3MAX1,C3MAX2
      PRINT 222,C2PLC,C2PPC,CSEL,CSEU
      PRINT 224,OV,RCSE,AD31,AD32,R3D
      PRINT 225,A4D,B4D,XK96,XK86,XKC,XKHAT
      PRINT 226
      DO 15 N=1,NSLUG
      15 PRINT 227,H,VIN(N),(CIN(I,N),I=1,NCOMP)
      PRINT 228
      PRINT 229,A11,A12,A21,A22
C
      PRINT 230,S1,S2,PERM1,PERM2,VIS1,VIS2
      PRINT 231,ICTL,ICTU
      PRINT 232,R1RW,S2RW,PIRW,P2RW,FF1,FF2
C
C***** SET UP INITIAL CONDITIONS *****
      DO 20 K=1,ICT
C
C PERMEABILITIES, FRACTIONAL FLOW, AND TOTAL MOBILITY
C PERM(K)=PERM1
C PERM(K)=PERM2
C PERM(K)=0.0
C
      FF(1,K)=FF1
      FF(2,K)=FF2
      FF(3,K)=0.0
C
      PHT(K)=PHTLU
C
C CONCENTRATIONS AND SATURATIONS
      DO 30 I=1,7
      DO 30 J=1,4
      30 C(I,J,K)=0.0
      C(1,1,K)=1.0
      C(2,2,K)=1.0
      C(5,1,K)=C511
      C(4,1,K)=CA11
C
      S(1,K)=S1
      S(2,K)=S2
      S(3,K)=1.0-S1-S2
      C(2,4,K)=S2

```



```

5      #WATER VISCOSITY          1X,F8.4,/,5X,
6      #OIL VISCOSITY           1X,F8.4,/)
231 FORMAT(5X,#PRESSURE DROP RECORDED BETWEEN#,I3,# AND#,I3)
232 FORMAT(/,5X,
1      #RESIDUAL WATER SATURATION 1X,F8.4,/,5X,
2      #RESIDUAL OIL SATURATION   1X,F8.4,/,5X,
3      #END POINT REL. PERM. FOR WATER 1X,F8.4,/,5X,
4      #END POINT REL. PERM. FOR OIL  1X,F8.4,/,5X,
5      #WATER FRACTIONAL FLOW      1X,F8.4,/,5X,
6      #OIL FRACTIONAL FLOW        1X,F8.4)
240 FORMAT(/5X,#POPE'S MODEL IS USED IF THREE PHASES APPEAR*)
241 FORMAT(/5X,#HIRASAKI'S MODEL IS USED IF THREE PHASES APPEAR*)
242 FORMAT(/5X,#MODIFIED HIRASAKI'S MODEL IS USED IF THREE PHASES APPEAR*)
243 FORMAT(/5X,#LAKE'S MODEL IS USED IF THREE PHASES APPEAR*)
      END

```

```

      SUBROUTINE OUTPUT
C-----
C DEPENDING ON HOW MUCH P.V. IS INJECTED, THIS SUBPROGRAM CALL SOME OF
C SUBROUTINES LISTED BELOW.
C MPRINT : PRINT PRODUCTION DURING THE TIME INCREMENT AND
C          SAVE VALUES FOR PLOT
C PROF : PRINT PROFILE
C PRFPLUT : PLOT PROFILE
C HISPLUT : PLOT HISTORY
C MATBAL : CALCULATE MATERIAL BALANCE ERROR AT THE TERMINATION
C          OF INJECTION
C EVERY 0.01 P.V. PRODUCTION IS PRINTED
C EVERY 0.25 P.V. PROFILE IS PRINTED AND PLOTTED
C WHEN SLUG CONCENTRATION INJECTED IS CHANGED, BOUNDARY CONDITION
C AT INJECTOR IS CHANGED IN THIS SUBPROGRAM.
C-----
C
C COMMON/HAIR/ISOLV, NSTOP, ID
C COMMON/NO/ICT1, ICT2, XICT, NCOMP
C COMMON/IN/VIN(10), CIN(7,10)
C COMMON/INJECT/INVP, VP, VT, VPI
C COMMON/SOL/C(1,0,02), S(3,02), FF(3,02), NPHASE(02)
C COMMON/SFHI01/DIMAX, ERR, YBIAS, IPASS
C COMMON/SFHI02/IEVA, DTOLD, RTEMAX, NREFJ, IANS
C COMMON/OUT/IN
C COMMON/DT/XDT(2002,1), YDT(2002,1), IPV
C
C DATA ISLUG, NSTOP, IPV/1,0,0/
C DATA VPH, VPP/0.01,0.25/
C
C ***** SAVE DTOLD FOR PLOT (SFHI-DISCRETE METHOD) *****
C IF (ISOLV.NE.1) GO TO 30
C IPV=IPV+1
C XDT(IPV,1)=VP
C YDT(IPV,1)=DTOLD
C 30 CONTINUE
C
C ***** CHECK WHETHER COMPLETION OF JOB *****
C IF ((VT-VP).LT.1.E-12) GO TO 10
C
C ***** CHANGE BOUNDARY CONDITION AT INJECTOR IF NECESSARY *****
C IF ((VIN(ISLUG)-VP).GT.1.E-12) GO TO 20
C ISLUG=ISLUG+1
C VPI=VIN(ISLUG)
C IPASS=0
C DO 25 I=1, NCOMP
C 25 C(I,1, ICT1)=CIN(I, ISLUG)
C 20 CONTINUE
C
C ***** PRINT HISTORY AND SAVE DATA FOR PLOT *****
C IF ((VPH-VP).GT.1.E-12) RETURN
C VPH=VPH+0.01
C CALL MPRINT
C
C ***** PRINT AND PLOT PROFILE *****
C IF ((VPP-VP).GT.1.E-12) RETURN
C VPP=VPP+0.25
C IN=0
C CALL PROF
C CALL PRFPLUT
C RETURN

```

```

C
C***** COMPLETION OF JOB *****
10 CONTINUE
  CALL HPRINT
  CALL PROF
  CALL PRFPLOT
  CALL HISPLOT
  CALL HISTRAL
  IF (ISOLV.EQ.1) PRINT 100, NREJ, IEVA
100 FORMAT(///15X, *NUMBER OF REJECTION **,13/
1      /15X, *NUMBER OF FUNCTION EVALUATION **,15)
  NSTOP=1
  RETURN
END

```

```

SUBROUTINE HPRINT
C-----
C THIS SUBROUTINE PRINT PRODUCTION HISTORY AND SAVE VALUES FOR PLOT
C-----
C
COMMON/NQ/ICT,ICT1,ICT2,YICT,NCOMP
COMMON/INJECT/HVP,VP,VT,VPT
COMMON/PRODIN/ER,P(7),PA,ZI(7),ZE(7),S2
COMMON/PPRES/PHTLU,PRESUM,ICTL,ICTH,PRES(40)
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1PW,S2RW,PHT(40)
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPMASE(42)
COMMON/SENIO2/IEVA,DOLD,RTMAX,NREJ,IAD3
COMMON/OUT/IN
COMMON/CPLT/CSMAX,C6MAX

C
COMMON/HIS1/YH3(S04,7),YH4(S04,7),FFP(S04,3),FFPP(S04,1)
COMMON/HIS2/XDH(S04,7),PREMAX,IPT

C
DATA PREMAX/1.0/
DATA IN,IPT/0.0/

C
C***** PRINT PRODUCTION HISTORY *****
  IN=IN+1
  IF (IN.GE.1) GO TO 10
  PRINT 220
  PRINT 230, VP, (P(I),I=1,7),ER,P0,PHT(1),PRESUM,IEVA
  DO 30 J=1,3
30 PRINT 240, J, (C(I,J,ICT2),I=1,7),S(J,ICT2)
  PRINT 260, (C(I,4,ICT2),I=1,7)
  GO TO 20
10 CONTINUE
  PRINT 250, VP, (P(I),I=1,7),ER,P0,PHT(1),PRESUM,IEVA
  DO 50 J=1,3
50 PRINT 240, J, (C(I,J,ICT2),I=1,7),S(J,ICT2)
  PRINT 260, (C(I,4,ICT2),I=1,7)
  IF (IN.GE.7) IN=0
  20 CONTINUE

C
C***** SAVE HISTORY DATA TO BE PLOTTED *****
  IPT=IPT+1
  DO 4000 I=1,7
  YH4(IPT,I)=C(I,4,ICT2)
  XDH(IPT,I)=VP
4000 CONTINUE
  YH4(IPT,3)=YH4(IPT,3)+S,0
  IF (CSMAX.GT.0.5) YH4(IPT,5)=YH4(IPT,5)/CSMAX
  IF (C6MAX.GT.0.5) YH4(IPT,6)=YH4(IPT,6)/C6MAX
  YH3(IPT,1)=C(3,3,ICT2)
  YH3(IPT,2)=C(3,3,ICT2)+C(7,3,ICT2)
  YH3(IPT,3)=YH3(IPT,2)+C(1,3,ICT2)
  YH3(IPT,4)=YH3(IPT,3)+C(2,3,ICT2)
  FFP(IPT,1)=S(2,ICT2)
  FFP(IPT,2)=S(2,ICT2)+S(3,ICT2)
  FFP(IPT,3)=ER
  FFP(IPT,1)=PRESUM
  IF (PRESUM.GT.0) PREMAX=PRESUM

C
C***** FORMAT *****
C
220 FORMAT (1H1)
230 FORMAT (/,3X,2HVP,7X,2HP1,5X,2HP2,5X,2HP3,5X,2HP4,5X,2HP5,5X,2HP6

```

```

1,5X,2HP7,1PX,2HP8,1PX,2HP8,15X,8HTOT,MOR,,5X,8HPRESSURE,5X,4HIEVA/
21X,F5,3,3X,7(1X,F6,4),5X,F6,4,AX,F6,4,11X,F6,4,AX,F6,4,5X,15//
35X,5HPHASE,6X,27MCONCENTRATION OF COMPONENTS,17X,9HPHASE CUT/
411X,2HC1,5X,2HC2,5X,2HC3,5X,2HC4,5X,2HC5,5X,2HC6,5X,2HC7)
24H FORMAT (6X,13,7(1X,F6,4),5X,F6,4,AX,F6,4)
25H FORMAT (//,1X,F5,3,3X,7(1X,F6,4),5X,F6,4,8X,F6,4,11X,F6,4,AX,F6,4
1,5X,15/)
26H FORMAT (6X,3H 4,7(1X,F6,4),2(5X,F6,4))
RETURN
END

```

```

SUBROUTINE PROF
C-----
C THIS SUPROUTINE PRINT PROFILE
C-----
C
C DIMENSION GRA1(40),GRA2(40),GRA3(40),XDL(40)
C
COMMON/NO/ICT,ICT1,ICT2,XICT,MCOMP
COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEDP
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/INJECT/DVP,VP,VT,VPI
COMMON/PERMC/PERM(3,40),SRED(3,40),SN(3,40),VIS(3,40)
COMMON/ADSORP/C3ADSS(40),C6ADSS(40),C6ADSS(40),CAHATS(40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTM
COMMON/CR/CRAOSS(40),CCB(42)
COMMON/PRESS/PHTLU,PRESUM,ICTL,ICTU,PRES(40)
COMMON/TRAP/DUMMY(8),PHT(40)
C
C LINE=0
C NPAGE=1CT/4
C NICT=NPAGE+4
C IF(NICT.LT,ICT)NPAGE=NPAGE+1
C
C DO 34 N=1,NPAGE
C N1=N-1
C N2=N+3
C IF(N2.GT,ICT)N2=ICT
C
C CALCULATE IFT AND DIMENSIONLESS DISTANCE FROM INJECTOR
C DO 40 I=N1,N2
C IF(NPHASE(I).EQ,3)GO TO 41
C IF(NPHASE(I).EQ,1)GO TO 42
C IF((C(3,4,I)+C(7,4,I)).LT,0.0001)GO TO 43
C IF(C(3,4,I).LE,0.0)GO TO 43
C IF(CSE(I).GE,CSEU,0.0,NPHASE(I).EQ,4)GO TO 44
C GRA2(I)=1.0,0.0*XIFT2(I)
C GRA1(I)=GRA3(I)=0.0
C GO TO 40
C 41 CONTINUE
C GRA1(I)=1.0,0.0*XIFT1(I)
C GRA2(I)=GRA3(I)=0.0
C GO TO 40
C 42 CONTINUE
C GRA1(I)=GRA2(I)=0.0
C GRA3(I)=1.0,0.0*XIFTM
C GO TO 40
C 43 CONTINUE
C GRA1(I)=1.0,0.0*XIFT1(I)
C GRA2(I)=1.0,0.0*XIFT2(I)
C GRA3(I)=1.0,0.0*XIFTM
C
C 44 XDL(I)=FLOAT(ICT-I+1)/FLOAT(ICT)
C
C PRINT COMPUTED RESULTS
C IF(LINE.EQ,1)GO TO 44
C PRINT 100,VP
C LINE=1
C GO TO 47

```

```

46 CONTINUE
LINE=9
PRINT 102
47 CONTINUE
PRINT 110, (XDL(I), NPHASE(I), I=N1, N2)
PRINT 120, (CSE(I), CSE(I), I=N1, N2)
PRINT 125, (PHT(K), PRES(K), K=N1, N2)
PRINT 130
PRINT 140, ((S(J, K), J=1, 3), K=N1, N2)
DO 50 I=1, NCOMP
50 PRINT 150, I, ((C(I, J, K), J=1, 4), K=N1, N2)
PRINT 160, ((SRED(J, K), J=1, 3), K=N1, N2)
PRINT 170, ((SN(J, K), J=1, 3), K=N1, N2)
PRINT 180, ((PERM(J, K), J=1, 3), K=N1, N2)
PRINT 190, ((VIS(J, K), J=1, 3), K=N1, N2)
PRINT 200, ((FF(J, K), J=1, 3), K=N1, N2)
PRINT 210, (GRA1(K), GRA2(K), GRA3(K), K=N1, N2)
PRINT 220, (C3ADSS(K), C4ADSS(K), C5ADSS(K), C6HATS(K), K=N1, N2)
PRINT 230, (CC8(K), C4ADSS(K), K=N1, N2)
30 CONTINUE
C
C***** FORMAT *****
C
100 FORMAT(1H1, 3X, *PROFILE AT *, F5.3, * P.V. INJECTED*/3X, 16(2H**)/)
102 FORMAT(1X, 131(1H-)/)
110 FORMAT(1X, 2HYD, 4X, 4(7X, F5.4, 2X, 8H(NPHASE=, 11, 1H), 6X))
120 FORMAT(1X, 8HCSE, CSEP, 2X, 4(5X, F7.4, 7X, F7.4, 4X))
125 FORMAT(1X, 10HMO, DELP, 4(5X, F7.4, 7X, F7.4, 4X))
130 FORMAT(1X, 5HNPASE, SY, 4(5X, 1H1, 6X, 1H2, 6X, 1H3, 5X, 5HTOTAL))
140 FORMAT(1X, 10HSATURATION, 4(2X, 3F7.4, 7X))
150 FORMAT(5X, 1HC, 11, 4X, 4(2X, 4F7.4))
160 FORMAT(1X, 10HRES, SAT, 4(2X, 3F7.4, 7X))
170 FORMAT(1X, 10HNO, SAT, 4(2X, 3F7.4, 7X))
180 FORMAT(1X, 10HREL, PERM, 4(2X, 3F7.4, 7X))
190 FORMAT(1X, 10HVISCOSITY, 4(2X, 3F7.4, 7X))
200 FORMAT(1X, 10HFRAC, FLOW, 4(2X, 3F7.4, 7X))
210 FORMAT(1X, 3HIFT, 7X, 4(6X, 2HMM, 7X, 2HMO, 7X, 2HNO, 4X)/
1 1X, 10H(DYNE/CM), 4(3X, 3F9.5))
220 FORMAT(11X, 4(5X, 2HC3, 5X, 2HC4, 5X, 2HC6, 3X, 6HC6HATS)/
1 1X, 10HADSORPTION, 4(2X, 4F7.4))
230 FORMAT(11X, 4(5X, 2HC4, 3X, 6HC5ADSS, 14X)/
1 1X, 7HCOMPLEX, 3X, 4(2X, 2F7.4, 14X))
RETURN
END

```

```

SURROUTINE PRFLOT
C-----
C THIS SURROUTINE MAKE PLOTS OF PROFILE
C-----
C
DIMENSION NPTS(7), LINTYP(7)
DIMENSION YS(42, 3), YC(42, 4), F81(42, 3), F82(42, 3), F83(42, 3)
DIMENSION Y3(42, 4), Y4(42, 7), YH(42, 2), XFI(42, 7), XDL(40), YFF(42, 3)
DIMENSION YCSE(42, 4), YIFT(42, 4)
C
COMMON/MAIN/ISOLV, NSTOP, ID
COMMON/NO/ICT1, ICT2, XICT, NCOMP
COMMON/CSE/CSE(42), CREL, CSEU, RCSE, CSEOP
COMMON/SOL/C(7, 4, 42), S(3, 42), FF(3, 42), NPHASE(42)
COMMON/INJECT/DVP, VP, VT, VPT
COMMON/PERM/PERM(3, 40), SRED(3, 40), SN(3, 40), VIS(3, 40)
COMMON/ADSORP/C3ADSS(40), C4ADSS(40), C5ADSS(40), C6HATS(40)
COMMON/XIFT/XIFT1(42), XIFT2(42), XIFT3(42), XIFTW
COMMON/CF/C6ADSS(40), CCF(42)
COMMON/PRES3/PHTLU, PRESUN, ICTL, ICTU, PRES(40)
COMMON/TRAP/DUMMY(A), PHT(40)
COMMON/CPLT/CSMAX, C6MAX
C
COMMON/PLTISZ/TEMP(4), XL, VL
COMMON/FXUSCL/XF, YF, XD, YD, IX, IY
COMMON/SYMBCC/ISYMCC(21)
COMMON/XYTYP/IXA, IYA
COMMON/LINMOD/LINMOD(21)
COMMON/XLABL/XLABEL(3), YLABEL(3), NXCHAR, NYCHAR, LABSID, LARTYP,
ISIZT, ISIZN, NOEC
COMMON/TITL/NTITLE, ITITL(5, 5)
C
C***** SAVE VALUES TO BE PLOTTED *****
DO 500 I=1, ICT
XDL(K)=XICT*FLOAT(ICT-K+1)
Y3(K, 1)=S(1, K)
Y3(K, 2)=S(1, K)+S(3, K)
Y3(K, 3)=Y3(K, 2)+S(2, K)
C
YH(K, 1)=PHT(K)/PHTLU
C
F81(K, 1)=SRED(1, K)
F82(K, 1)=SRED(2, K)
F83(K, 1)=SRED(3, K)
F81(K, 2)=S(1, K)
F82(K, 2)=S(2, K)
F83(K, 2)=S(3, K)
F81(K, 3)=1.-SRED(2, K)-SRED(3, K)
F82(K, 3)=1.-SRED(1, K)-SRED(3, K)
F83(K, 3)=1.-SRED(1, K)-SRED(2, K)
C
YFF(K, 1)=FF(1, K)
YFF(K, 2)=FF(1, K)+FF(3, K)
YFF(K, 3)=YFF(K, 2)+FF(2, K)
C
YC(K, 1)=C(3, 4, K)
YC(K, 2)=C(3, 4, K)+C(7, 4, K)
YC(K, 3)=YC(K, 2)+C(1, 4, K)
YC(K, 4)=YC(K, 3)+C(2, 4, K)
C
YCSE(K, 1)=CSE(K)

```

```

      YCSE(K,2)=CSEOP
      YCSE(K,3)=CSEFL
      YCSE(K,4)=CSEU
C
      YIFT(K,1)=YIFT1(K)
      YIFT(K,2)=YIFT2(K)
C
      Y3(K,1)=C(3,3,K)
      Y3(K,2)=C(3,3,K)+C(7,3,K)
      Y3(K,3)=Y3(K,2)+C(1,3,K)
      Y3(K,4)=Y3(K,3)+C(2,3,K)
C
      DO 5002 I=1,NCOMP
      XU1(K,I)=XDL(K)
      Y4(K,1)=C(1,4,K)
5002 CONTINUE
      Y4(K,3)=Y4(K,3)+5.0
      IF(CSMAX.GT.0.0)Y4(K,5)=Y4(K,5)/CSMAX
      IF(C6MAX.GT.0.0)Y4(K,6)=Y4(K,6)/C6MAX
5001 CONTINUE
C***** PLOT PROFILE (PRINTER OR ZETA PLOT) *****
      IPLT=1
      IX=0
      IY=0
      DO 6000 I=1,NCOMP
      LINTYP(I)=3
      ISYMCC(I)=1
6000 NPTS(I)=ICT
      NMAX2=02
      IF((0.6-VP).GT.0.0)IPLT=3
      IF(IPLT.NE.3)GO TO 6100
      IX=IY=2
      CALL PLOTS(0,0,SLPLOT)
6100 CONTINUE
      NTITLE=2
      XL=YL=5.0
      XU=YO=0.2
      XF=YF=0.0
      LABTYP=0
      NDEC=1
      ITITL(1,1)=9HPPROFILES
      ITITL(2,1)=4HAT
      ENCODE(0,1000,ITITL(3,1))VP
      ITITL(4,1)=6H P.V.
      ITITL(1,2)=ITITL(2,2)=10H
      YLABEL(2)=YLABEL(3)=10H
      XLABEL(1)=10HFRACTIONAL
      YLABEL(2)=10H DISTANCE
      NXCHAR=20
C
C TOTAL CONCENTRATION
      YLABEL(1)=10HTOTAL CONC
      YLABEL(2)=10HCONCENTRATION
      NYCHAR=20
      NVECT=NCOMP
      CALL THEPLT(XD1,Y4,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C TOTAL COMPOSITION
      YLABEL(1)=10HTOTAL COMP
      YLABEL(2)=7HPOSITION
      NYCHAR=17

```

```

      NVECT=0
      CALL THEPLT(XD1,YC,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C SALINITY
      YLABEL(1)=9HSALINITY
      YLABEL(2)=10H
      NYCHAR=0
      YD=2.+CSEOP/5.0
      LINTYP(2)=LINTYP(3)=LINTYP(4)=0
      LINMOD(2)=6
      LINMOD(3)=4
      LINMOD(4)=4
      NVECT=4
      CALL THEPLT(XD1,YCSE,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
      YD=0.2
      DO 6200 I=2,4
      LINTYP(I)=3
6200 LINMOD(I)=2
C
C PHASE SATURATIONS
      YLABEL(1)=10HMPHASE SATU
      YLABEL(2)=6HURATION
      NYCHAR=16
      NVECT=3
      CALL THEPLT(XD1,Y5,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C MICROEMULSION PHASE CONCENTRATION
      YLABEL(1)=10HCONCENTRAT
      YLABEL(2)=4HIONS
      NYCHAR=14
      ITITL(1,2)=10HMICROEMULS
      ITITL(2,2)=10HION PHASE
      NVECT=4
      CALL THEPLT(XD1,Y3,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C INTERFACIAL TENSION
      YLABEL(1)=10HINTERFACIA
      YLABEL(2)=10HL TENSION
      YLABEL(3)=10H(DVNE/CM)
      NYCHAR=30
      ITITL(1,2)=ITITL(2,2)=10H
      NVECT=2
      IY=0
      IYA=0
      CALL THEPLT(XD1,YIFT,IPLT,NVECT,NPTS,NMAX2,LINTYP)
      YLABEL(3)=10H
      IYA=1
C
      IF(IPLT.NE.3)GO TO 7100
      CALL PLOT(XOUM,YOUM,999)
      CALL HELFASE(IN)
      CALL PLOTS(X,0,SLPLOT)
      IY=2
      YD=0.2
      YF=0.0
7100 CONTINUE
C
C FRACTIONAL FLOW
      YLABEL(1)=10HFRACTIONAL
      YLABEL(2)=5H FLOW
      NYCHAR=15

```

```

      NVECT=3
      CALL THEPLT(XD1,YFF,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C   RESIDUAL AND PHASE SATURATION
      YLABEL(1)=10HMPHASE AND
      YLABEL(2)=10HRESIDUAL S
      YLABEL(3)=9HATURATION
      NYCHAR=20
      ITITL(1,2)=10HMICROFMLS
      ITITL(2,2)=10HION PHASE
      NVECT=3
      CALL THEPLT(XD1,FS3,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
      ITITL(1,2)=10H AQUEOUS
      ITITL(2,2)=10HMPHASE
      NVECT=3
      CALL THEPLT(XD1,FS1,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
      ITITL(1,2)=10H OLEIC
      NVECT=3
      CALL THEPLT(XD1,FS2,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
C   TOTAL RELATIVE MOBILITY
      ITITL(1,2)=ITITL(2,2)=10H
      YLABEL(1)=10HTOTAL REL
      YLABEL(2)=10HTATIVE MOBIL
      YLABEL(3)=10HTITY
      NYCHAR=23
      NVECT=1
      IY=8
      CALL THEPLT(XD1,YM,IPLT,NVECT,NPTS,NMAX2,LINTYP)
C
      IF(IPLT.NE.3)GO TO 7000
      CALL PLOT(XDUM,YDUM,999)
      CALL RELEASE(ID)
7000 CONTINUE
10000 FORMAT(F8.2)
      RETURN
      END

```

```

      SUBROUTINE HISPLT
C-----
C   THIS SUBPROGRAM PRODUCES PLOTS OF HISTORY.
C-----
C
      DIMENSION NPT(7),LINTYP(4)
C
      COMMON/MAIN/ISOLV,NSTOP,TD
      COMMON/HIS1/YH1(504,7),YH2(504,7),FFP(504,3),FFPP(504,1)
      COMMON/HIS2/XDH(504,7),PREMAX,IPT
      COMMON/DI/XDT(2002,1),YDT(2002,1),IPV
      COMMON/SFHID1/DTHAX,ERR,YBIAS,IPASS
      COMMON/INJECT/DVP,VP,VY,VPI
      COMMON/PRODIN/ER,P(7),PA,ZI(7),ZE(7),S2
C
      COMMON/PLTSIZ/TEMP(4),XL,YL
      COMMON/FXNSCL/XF,YF,XD,YD,IX,IY
      COMMON/SYMBCC/ISYBCC(21)
      COMMON/AXTYP/IXA,IYA
      COMMON/LINMOD/LINMOD(21)
      COMMON/AXLABL/XLABL(3),YLABL(3),NXCHAR,NYCHAR,LARSTO,LARTYP,
      ISIZT,SIZN,NDEC
      COMMON/TITL /NTITLE,ITITL(5,5)
C***** PLOT HISTORY (PRINTER PLOT) *****
      IPLT=1
      NMAX3=504
      DO 4002 I=1,7
      LINTYP(I)=3
4002 NPT(I)=IPT
      IX=8
      XLABL(1)=10HP.V. INJEC
      XLABL(2)=10HTE
      YLABL(1)=10H
      YLABL(2)=10HCONCENTRAT
      YLABL(3)=10HIONS
      NXCHAR=13
      NYCHAR=30
      ITITL(1,1)=10HHISTORIES
      ITITL(2,1)=10HSHUP TO
      ENCODE(4,100H,ITITL(3,1)) VP
      NVECT=4
      CALL THEPLT(XDH,YH3,IPLT,NVECT,NPT,NMAX3,LINTYP)
      NVECT=7
      CALL THEPLT(XDH,YH4,IPLT,NVECT,NPT,NMAX3,LINTYP)
      IF(S2.LE.0)GO TO 900
      NVECT=3
      YLABL(1)=10HMPHASE CUT
      YLABL(2)=10HNOIL REC
      YLABL(3)=10HIVERY
      CALL THEPLT(XDH,FFP,IPLT,NVECT,NPT,NMAX3,LINTYP)
900 CONTINUE
      YLABL(1)=10H
      YLABL(2)=10HREL PRESS
      YLABL(3)=10HNDROP
      ITITL(1,1)=10H
      ITITL(2,1)=10H
      ITITL(3,1)=10H
      ITITL(4,1)=10H
      NVECT=1
      CALL THEPLT(XDH,FFPP,IPLT,NVECT,NPT,NMAX3,LINTYP)
      YLABL(2)=10HTIME STEP

```



```

        YLABEL(3)=1000000
        NMAX3=2002
        NPT(1)=IPV
        CALL THEPLT(XDT,YDT,IPLT,NVECT,NPT,NMAX3,LINTYP)
C
C***** PLOT HISTORY (ZETA PLOT) *****
        CALL PLOTS(R,0,SLPLOT)
        NTITLE=-1
        IPLT=3
        IX=2
        IY=2
        XL=5,AVT
        YL=5,
        XD=0,2
        YD=0,2
        XF=0,
        YF=0,
        NMAX3=5000
        DO 200 I=1,7
200    NPT(I)=IPT
        DO 100 I=1,8
        LINTYP(I)=10
100    ISYMCC(I)=J
        XLABEL(1)=1000,V. INJEC
        XLABEL(2)=3000
        YLABEL(2)=1000 PHASE CON
        ITITL(2,1)=5000 TO
        ENCODE(4,1000,ITITL(1,1)) VP
        ITITL(4,1)=1000 P.V. INJ
        ITITL(5,1)=100000
        NVECT=7
        YLABEL(1)=1000 MICRO.
        YLABEL(3)=1000
        ITITL(1,1)=1000 HISTORIES
        CALL THEPLT(XDH,YH3,IPLT,NVECT,NPT,NMAX3,LINTYP)
        YLABEL(1)=1000 TOTAL
        YLABEL(2)=1000 CONCENTRA
        YLABEL(3)=1000000
        CALL THEPLT(XDH,YH4,IPLT,NVECT,NPT,NMAX3,LINTYP)
C
C PLOT PHASE CUTS AND CUMULATIVE OIL RECOVERY
        IF(S2,LE,0,0)GO TO 910
        NVECT=3
        YLABEL(1)=1000 PHASE CUT
        YLABEL(2)=100000 OIL REC
        YLABEL(3)=1000000
        CALL THEPLT(XDH,FFP,IPLT,NVECT,NPT,NMAX3,LINTYP)
910 CONTINUE
C
C PLOT RELATIVE PRESSURE DROP
        IX=2
        IY=0
        NVECT=1
        YLABEL(1)=1000 RELATIVE
        YLABEL(2)=1000 PRESSURE
        YLABEL(3)=1000000
        IF(PREMAX,LE,5,0) IY=2
        IF(PREMAX,LE,5,0) YD=1,0
        CALL THEPLT(XDH,FFPP,IPLT,NVECT,NPT,NMAX3,LINTYP)
        IF(ISOVL,EO,0)GO TO 9100
C
C PLOT TIME STEP SIZE

```

```

        YLABEL(1)=1000 TIME
        YLABEL(2)=1000 STEP SIZE
        YLABEL(3)=1000
        NPT(1)=IPV
        NMAX3=2002
        IY=2
        YF=0,0
        YD=0,002
        IF(DTMAX,GT,0,01)YD=0,000
        IF(DTMAX,GT,0,02)YD=0,001
        IF(DTMAX,GT,0,05)YD=0,002
        LINTYP(1)=0
        CALL THEPLT(XDT,YDT,3,1,NPT,NMAX3,LINTYP)
9100 CONTINUE
        CALL PLOT(XDUM,YDUM,999)
        CALL RELEASE(ID)
1000 FORMAT(F4,2)
        RETURN
        END

```

```

SUBROUTINE SOLVE
C-----
C THIS SUBPROGRAM SOLVES CONTINUITY EQUATIONS,
C FULLY-DISCRETE FORWARD EULER IS USED,
C CUMULATIVE PRODUCTION AND RELATIVE PRESSURE DROP ARE ALSO CALCULATED.
C-----
COMMON/NO/ICT,ICT1,ICT2,XICT,NCOMP
COMMON/SYSTEM/UT,AHPRM,PHI,EPHI3,EPHI4,DISPJ(4)
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEOP
COMMON/A3D/A3DS(40),A3S1,A3S2
COMMON/CHEMAD/C3PH,A3D,B3D
COMMON/POLYAD/C4PH,A3D,B4D
COMMON/HMR/XKC,XK16,XK16,XKHAT,BV
COMMON/ADSORP/C3ADSS(42),C4ADSS(40),C6ADSS(40),C6HAT8(40)
COMMON/CR/C8ADSS(40),CC8(42)
COMMON/INJECT/DVP,VP,VT,VP1
COMMON/PFOUIN/ER,P(7),P8,ZI(7),ZE(7),S2
COMMON/TRAP/DUMMY(R),PHT(40)
COMMON/PRESS/PHTLI,PRESUM,ICTL,ICTU,PRES(40)
COMMON/ION/FFDV,DC3,K

C
C***** CONCENTRATIONS IN PRODUCTION *****
DO 357 I=1,NCOMP
  C(I,4,ICT2)=0.0
DO 357 J=1,3
  C(I,4,ICT2)=C(I,4,ICT1)+C(I,J,1)*FF(J,1)
357 CONTINUE
CSE(ICT2)=C(5,4,ICT2)-C(6,4,ICT2)+RCSE+C(6,4,ICT2)/
  IC(1,4,ICT2)

C
C***** CHECK TIME STEP SIZE *****
FFDV=FFDV
DVP=DVP
VP=VP+DVP
IF((VP-VP1).LT.2.0)GO TO 12
DVP=VP+VP1-VP
VP=VP1
FFDV=FFDV+DVP/DVP
12 CONTINUE

C
C***** CUMULATIVE PRODUCTION *****
DO 13 I=1,NCOMP
  P(I)=P(I)+C(I,4,ICT2)*DVP
  PB=FF(1,1)+CC8(1)+DVP+PB
  EN=P(2)/S2

C
C***** REFININE TOTAL COMPOSITION BY INCLUDING ADSORPTION *****
DO 355 K=1,ICT
  X=1.-C3ADSS(K)
  C(1,4,K)=C(1,4,K)*X
  C(2,4,K)=C(2,4,K)*X
  C(3,4,K)=C(3,4,K)*X+C3ADSS(K)
  C(7,4,K)=C(7,4,K)*X
355 CONTINUE

C
C***** SOLVE CONTINUITY EQUATION *****
DO 29 KK=1,ICT
  K=ICT+1-KK
  DO 34 I=1,NCOMP
    FFUN=GFUN+I,K

```

```

    DU 25 J=1,3
    IF(K.EQ. 1) GO TO 40
C
C MATERIAL TRANSPORT BY DISPERSION
FFUN=FFUN+DISPJ(J)*(FF(J,K+1)-C(I,J,K+1)-C(I,J,K))-
  IFF(J,K)*(C(I,J,K)-C(I,J,K-1)))*FLOAT(ICT)
GO TO 25
40 FFUN=FFUN+DISPJ(J)*(FF(J,2)*(C(I,J,2)-C(I,J,1))-
  IFF(J,1)*(C(I,J,1)-C(I,J,2)))*FLOAT(ICT)
C
C MATERIAL TRANSPORT BY CONVECTION
25 GFUN=GFUN-(FF(J,K+1)*C(I,J,K+1)-FF(J,K)*C(I,J,K))
  EPHI=I,K
C
C INACCESSIBLE PORE VOLUMES TO SURFACTANT AND/OR POLYMER
IF(I.EQ. 3) EPHI=EPHI3
IF(I.EQ. 4) EPHI=EPHI4
C(I,4,K)=C(I,4,K)-FFDV*(GFUN-FFUN)/EPHI
IF(C(I,4,K).GE.-1.E-5)GO TO 999
PRINT 999,I,K,C(I,4,K)
997 FORMAT(//10X,'NEGATIVE CONCENTRATION OCCURED AT SUBPROGRAM SOLVE*/
  1 /15X,2HC(,11,3H,4,,12,3H) ',E15,7)
DO 910 KMG=1,ICT
910 PRINT 998,KMG,(C(ING,4,KMG),ING=1,NCOMP)
998 FORMAT(10X,I3,7G10,6)
STOP
900 CONTINUE
30 CONTINUE
C
C***** ION EXCHANGE *****
IF(QV,GE.0.0001)CALL IONCNR
C
C***** CHEMICAL ADSORPTION *****
CSE(K)=C(5,4,K)-C(6,4,K)+RCSE+C(6,4,K)/C(1,4,K)
A3D=A3S1+A3S2+CSE(K)
IF(C(3,4,K).LT.1.0E-4)GO TO 28
IF(A3D.LT. 1.0E-4) GO TO 28
C3PH=MAX1(C(3,1,K),C(3,2,K),C(3,3,K))
IF(NPHASE(K).EQ.1)C3PH=C(3,4,K)
CALL CHEMADH(C(3,4,K),C3ADSS(K),A3DS(K))
C
C***** REFININE TOTAL COMPOSITION BY EXCLUDING ADSORPTION *****
28 CTOT=C(1,4,K)+C(2,4,K)+C(3,4,K)+C(7,4,K)
  C(1,4,K)=C(1,4,K)/CTOT
  C(2,4,K)=C(2,4,K)/CTOT
  C(3,4,K)=C(3,4,K)/CTOT
  C(7,4,K)=C(7,4,K)/CTOT
C
C
IF(C(4,4,K).LT.1.0E-4)GO TO 29
IF(A4D.LT. 1.0E-4) GO TO 29
C
C***** POLYMER ADSORPTION *****
C4PH=MAX1(C(4,1,K),C(4,2,K),C(4,3,K))
IF(NPHASE(K).EQ.1)C4PH=C(4,4,K)
CALL POLYADH(C(4,4,K),C4ADSS(K))
29 CONTINUE
C
C***** PHASE CONCENTRATIONS AND PROPERTIES *****
CALL PROPTY
C
C***** RELATIVE PRESSURE DROP *****
PRES=I,K

```

```

DO 103 K=1,ICT
PRES(K)=PHTLU/PHT(K)
100 PRESUM=PRESUM+PRES(K)
PRESUM=PRESUM/FL0AT(1CTU-1CTL+1)
RETURN
END

```

```

SUBROUTINE SOLVE1
C-----
C THIS SUBPROGRAM SOLVES CONTINUITY EQUATIONS.
C SEMI-DISCRETE METHOD WITH NUMERICAL O.D.E. INTEGRATOR IS USED.
C O.D.E. INTEGRATOR IS SELECTED ACCORDING TO ISEM AS FOLLOWS:
C ISEM #1 : RK12
C #2 : RK1
C #3 : OGEAR
C-----
C DIMENSION CC(200),IAK(200),WK(16500),FFOLD(3),COLD(7,3)
C COMMON/GEAR/DUM(52),I0UM(30)
C
C COMMON/SEMI01/DIMAX,ERR,YRIAS,IPASS
C COMMON/SEMI02/IEVA,DTOLD,RTENAX,NREJ,IAN3
C COMMON/SEMI03/NEQ,ISEM,XEND,METH,MITER
C COMMON/NO/ICT1,ICT2,XICT,NCOMP
C COMMON/SOL/C(7,0,02),S(3,02),FF(3,02),HPHASE(02)
C COMMON/INJECT/DVP,VP,VT,VPI
C COMMON/PRDIN/ER,P(7),PA,ZI(7),ZE(7),S2
C COMMON/CSE/CSE(02),CSEL,CSEU,RCSE,CSEOP
C COMMON/TRAP/DUMMY(A),PHT(00)
C COMMON/PRESS/PHTLU,PRESUM,ICTL,ICTU,PRES(00)
C COMMON/CR/C8AUSS(00),CCR(02)
C EXTERNAL DER,FCNJ
C DATA IPASSOL,IPASS,INDEX/0,0,1/
C
C***** REARRANGE TOTAL CONCENTRATION ARRAY ONLY AT THE FIRST TIME ***
IF(IPASSOL.NE.0)GO TO 10
IPASSOL=1
DO 20 K=1,ICT
DO 20 I=1,NCOMP
I1=(K-1)*NCOMP+1
20 CC(I1)=C(I,0,K)
10 CONTINUE
C
C***** COMPOSITION OF PRODUCTION *****
DO 32 I=1,NCOMP
C(I,0,ICT2)=0.0
DO 32 J=1,3
C(I,0,ICT2)=C(I,0,ICT2)+C(I,J,1)*FF(J,1)
32 CONTINUE
CSE(1CT2)=(C(5,0,ICT2)-C(6,0,ICT2)+RCSE+C(6,0,ICT2))/
IC(1,0,ICT2)
C
C***** SAVE OLD VALUES FOR CALCULATION OF CUMULATIVE PRODUCTION *****
CCOLD=CCR(1)
DO 3A J=1,3
3A FFOLD(J)=FF(J,1)
C
C***** OBTAIN SOLUTION WITH SEMI-DISCRETE METHOD *****
IF(ISEM.NE.3)GO TO 60
C OGEAR IS USED
IF(1INDEX.EQ.1)DVP=0.3701
IF(DVP.GE.(VPI-VP))IYOFX=2
DUM(0)=DIMAX
CALL OGEAR(NEQ,DER,FCNJ,VP,DVP,CC,XEND,ERR,METH,MITER,INDEX,
I IAK,WK,IER)
XEND=VPI

```

```

INDEX=3
IF((VPI-VP).GT.1.E-12)GO TO 78
INDEX=1
XEND=XFND+0.0001
74 CONTINUE
PRINT 90H,VP,DVP,DIUM(8),IDUM(6),IEVA,IDUM(7),IDUM(9),INDEX
903 FURMAT(8X,4VP,DVP,HUSED,ORNER,IEVA,NSTEP,NJE,INDEX*,5X,3615,5,516)
GO TO 62
C
C RK12 OR RK1 IS USED
60 CONTINUE
IF(DVP.LT.1.E-6)DVP=0.001
IF(DVP.GT.(VPI-VP))DVP=VPI-VP
IF(ISEM.EQ.1)CALL RK12(VP,CC,NEQ,DVP)
IF(ISEM.EQ.2)CALL RK1(VP,CC,NEQ,DVP)
62 CONTINUE
C
C***** RELATIVE PRESSURE DROP AND TOTAL PRESSURE DROP *****
PRESUM=0.0
DO 40 K=1,ICT
PRES(K)=PHTLU/PHT(K)
40 PRESUM=PRESUM+PRES(K)
PRESUM=PRESUM/FLQAT(ICTU-ICTL+1)
C
C***** CUMULATIVE PRODUCTION *****
IF(ISEM.EQ.3)DTOLD=DVP
DO 54 I=1,NCOMP
54 P(I)=P(I)+DTOLD*C(I,4,ICT2)
P8=FFOLD(1)*CCOLD*DTOLD+P8
ENR(2)/S2
RETURN
END

```

```

SUBROUTINE FCN1(N,X,Y,PD)
C-----
C THIS IS A DUMMY PROGRAM NEEDED WHEN DGEAR (ISOLV=1,ISEM=3) IS USED.
C-----
RETURN
END

```

```

SUBROUTINE RK12(T,X,N,DT)
-----
C THIS IS AN O.D.E INTEGRATOR WITH STEP SIZE CONTROL.
C RUNGE-KUTTA-FEHLBERG ALGORITHM OF FIRST AND SECOND ORDER IS USED.
C COMPOSITION OF PRODUCTION IS CALCULATED IN THIS SUBPROGRAM.
-----
C
COMMON/SEMI01/DTMAX,ERR,YBIAS,IPASS
COMMON/SEMI02/IEVA,DTOLD,RTEMAX,NREJ,IADS
COMMON/H0/ICT,ICT1,ICT2,XICT,NCOMP
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
REAL X(1),Y(260),F1(260),F2(260),F3(260)
REAL PP1(7),PP2(7)
DATA NREJ/0/

C
IF (IPASS .NE. 0) GO TO 4
IPASS = 1
H21 = A2 = 1./2.
C1 = H31 = 1./256.
C2 = H32 = 255./256.
D1 = CH1 = CH3 = 1./512.
CH2 = 255./256.
PCT = .4
EBIAS = 1.E-12
PRINT 100,PCT
100 FORMAT(//1X,'PCT =',F5.3)
CALL DER(N,T,X,F1)
4 CONTINUE
IADS=0

C
DO 20 I=1,NCOMP
PP1(I)=0.0
DO 20 J=1,3
20 PP1(I)=PP1(I)+FF(J,I)*C(I,J,1)

C
7 CONTINUE
TY = T+A2*DT
NR21 = R21+DT
DO 2 I = 1,N
2 Y(I) = X(I)+NR21*F1(I)
CALL DER(N,TY,Y,F2)

C
DO 30 I=1,NCOMP
PP2(I)=0.0
DO 30 J=1,3
30 PP2(I)=PP2(I)+FF(J,I)*C(I,J,1)
30 C(I,5,ICT2)=(PP1(I)+255.*PP2(I))/256.

C
TY = T+DT
NR31 = R31+DT
NR32 = R32+DT
DO 3 I = 1,N
3 Y(I) = X(I)+NR31*F1(I)+NR32*F2(I)
CALL DER(N,TY,Y,F3)
DO1 = 01+DT
RTEMAX = EBIAS
DO 4 I = 1,N
4 ATE = ABS(F1(I)-F3(I))+DO1
AY = ABS(Y(I))
IF (AY .LT. YBIAS) AY = YBIAS
DER = ATE/AY

```

```

IF (RER .GT. RTEMAX) RTEMAX=RER
4 CONTINUE
DTOLD = DT
DT=DTOLD+PCT*(ERR/RTEMAX)*.5
DT = AMIN1 (DT,DTMAX)
IF (RTEMAX .LE. ERR) GO TO 7
DT = DT*.9
NREJ=NREJ+1
IADS=2
GO TO 9
7 T = T+DTOLD
DO 15 I = 1,N
F1(I) = F3(I)
15 X(I) = Y(I)
RETURN
END

```

```

      SUBROUTINE RK1(T,X,N,DT)
C-----
C THIS IS AN O.D.E. INTEGRATOR WITH STEP SIZE CONTROL.
C RUNGE-KUTTA METHOD OF FIRST AND SECOND ORDER ARE USED.
C-----
C
      COMMON/SEMI01/DTMAX,ERR,YBIAS,IPASS
      COMMON/SEMI02/IEVA,DTOLD,RTEMAX,NREJ,IADS
      COMMON/HO/ICT,ICT1,ICT2,XICT,NCOMP
      REAL X(1),Y(200),F1(200),F2(200),F3(200)
      DATA NREJ/0/

C
      IF (IPASS,NE,0) GO TO A
      IPASS = 1
      H21=H2=1.
      C1=1.
      CH1=CH2=1./2.
      D1=1./2.
      PCT = .8
      EBIAS = 1.E-12
      PRINT 100,PCT
100 FORMAT(//10X,*PCT =*,F4,3)
      CALL DER(N,T,X,F1)
      A CONTINUE
      IADS=0
      7 CONTINUE
      TY = T+A2*DT
      H21 = H21+DT
      DO 2 I = 1,N
      2 Y(I) = X(I)+H21*F1(I)
      CALL DER(N,TY,Y,F2)
      DD1 = D1+DT
      RTEMAX = F1(IAS)
      DO 4 I = 1,N
      ATE = ABS(F1(I)-F2(I))*DD1
      AY = ABS(Y(I))
      IF (AY,LT,YBIAS) AY = YBIAS
      HER = ATE/AY
      IF (HER,GT,RTEMAX) RTEMAX=HER
      4 CONTINUE
      DTOLD = DT
      DT=DTOLD+PCT*(ERR/RTEMAX)*.5
      DT = AMIN1 (DT,DTMAX)
      IF (RTEMAX,LE,ERR) GO TO 7
      DT = DT+.9
      NREJ=NREJ+1
      IADS=2
      GO TO 9
      7 T = T+DTOLD
      DO 15 I = 1,N
      F1(I) = F2(I)
15 X(I) = Y(I)
      RETURN
      END

```

```

      SUBROUTINE DER(N,T,Y,DY)
C-----
C THIS SUBROUTINE CALCULATES CHANGE IN CONCENTRATION AS DERIVATIVE
C WITH RESPECT TO TIME
C-----
C
      DIMENSION DY(200),Y(200),FFUN(7,40),GFIIN(7,40)
      DIMENSION C3ADOLD(40),A3DOLD(40),C4ADOLD(40),C3PHO(40),C4PHO(40)

C
      COMMON/SEMI02/IEVA,DTOLD,RTEMAX,NREJ,IADS
      COMMON/HO/ICT,ICT1,ICT2,XICT,NCOMP
      COMMON/SYSTEM/UT,AMPERN,PHI,EPHI1,EPHI2,DISPJ(4)
      COMMON/SOL/C(7,4,42),S(3,42),FF(1,42),NPHASE(42)
      COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEOP
      COMMON/A3D/A3DS(40),A331,A332
      COMMON/CHEMAD/C3PH,A3D,B3D
      COMMON/POLYAD/C4PH,A4D,B4D
      COMMON/HMR/XKC,XK96,XK9A,XKHAT,QV
      COMMON/ADSORP/C3ADSS(40),C4ADSS(40),C6ADSS(40),C6HATS(40)
      COMMON/CP/C6ADSS(40),CCA(42)
      DATA IEVA/0/
      DATA IPASDER,IADS/0,0/

C
      IEVA=IEVA+1

C
C***** SAVE VALUES WHICH SHOULD BE USED IN CASE OF REJECTION *****
      IF (IADS,NE,0) GO TO 50
      DO 52 K=1,ICT
      C3ADOLD(K)=C3ADSS(K)
      A3DOLD(K)=A3DS(K)
      C4ADOLD(K)=C4ADSS(K)
52 CONTINUE
50 CONTINUE
      IF (IPASDER,ER,0) GO TO 30

C
C***** RESET VALUES IN CASE OF REJECTION *****
      IF (IADS,NE,2) GO TO 56
      DO 54 K=1,ICT
      C3ADSS(K)=C3ADOLD(K)
      A3DS(K)=A3DOLD(K)
      C4ADSS(K)=C4ADOLD(K)
54 CONTINUE
56 CONTINUE

C
C***** RESET TOTAL COMPOSITION FROM Y'S SENT FROM RK12 *****
      DO 24 K=1,ICT
      DO 14 I=1,NCOMP
      II=(K-1)*NCOMP+1
      C(I,4,K)=Y(I1)
      IF (C(I,4,K),GT,-1.E-5) GO TO 10
      PRINT 100,I,K,C(I,4,K)
100 FORMAT(//5X,*NEGATIVE CONCENTRATION OCCURED IN SUBPROGRAM DER//
1 AY,PHC(I,3H,4.,12,3H) =,E15,7//)
      CALL PROF
      CALL PRFPLNT
      STOP
10 CONTINUE

C
C***** ION EXCHANGE *****
      IF (QV,GE,N,NONH) CALL IONXNG

C
C***** CHEMICAL ADSORPTION *****
      CSE(N)=(C(5,4,K)-C(6,4,K)+RCSE+C(6,4,K))/C(1,4,K)

```

```

      A3D=A031+A032+CSE(K)
      IF(C(3,4,K),LT,1,PE=0)GO TO 28
      IF(A3D,LT,1,PE=0) GO TO 28
      C3PH=AMAX1(C(3,1,K),C(3,2,K),C(3,3,K))
      IF(NPHASE(K),EQ,1)C3PH=C(3,4,K)
      IF(IADS,EQ,0)C3PHO(K)=C3PH
      IF(IADS,EQ,2)C3PH=C3PHO(K)
      CALL CHEHADN(C(3,4,K),C3ADSS(K),A3DS(K))

C ***** CHANGE DEFINITION OF TOTAL COMPOSITION
C ***** EXCLUDING ADSORBED AMOUNT OF SURFACTANT *****
28 CTOT=C(1,4,K)+C(2,4,K)+C(3,4,K)+C(7,4,K)
   C(1,4,K)=C(1,4,K)/CTOT
   C(2,4,K)=C(2,4,K)/CTOT
   C(3,4,K)=C(3,4,K)/CTOT
   C(7,4,K)=C(7,4,K)/CTOT

C ***** POLYMER ADSORPTION *****
   IF(C(4,4,K),LT,1,PF=0)GO TO 28
   IF(A0D,LT,1,PF=0) GO TO 28
   C(4,4,K)=C(4,4,K)-C4ADSS(K)
   C4PH=AMAX1(C(4,1,K),C(4,2,K),C(4,3,K))
   IF(NPHASE(K),EQ,1)C4PH=C(4,4,K)
   IF(IADS,EQ,0)C4PHO(K)=C4PH
   IF(IADS,EQ,2)C4PH=C4PHO(K)
   CALL POLYADN(C(4,4,K),C4ADSS(K))

29 CONTINUE
C ***** NEW PHASE COMPOSITIONS AND PROPERTIES *****
   CALL PROPRTY

C 30 CONTINUE
   IPASDEP=1

C ***** CALCULATE DERIVATIVES *****
   DO 29 JK=1,ICT
     K=ICT+1-TK
     DO 36 I=1,NCOMP
       FFUN(I,K)=GFUN(I,K)+0.0
     DO 25 J=1,3
       IF(K,EQ,1) GO TO 40
C MATERIAL TRANSPORT BY DISPERSION
       FFUN(I,K)=FFUN(I,K)-DISPJ(J)*(FF(J,K+1)*(C(I,J,K+1)-C(I,J,K))-
       IF(J,K)*(C(I,J,K)-C(I,J,K-1)))*FLOAT(ICT)**2
       GO TO 25
40 FFUN(I,K)=FFUN(I,K)-DISPJ(J)*(FF(J,2)*(C(I,J,2)-C(I,J,1))-
       IF(J,1)*(C(I,J,1)-C(I,J,0)))*FLOAT(ICT)**2
C MATERIAL TRANSPORT BY CONVECTION
25 GFUN(I,K)=GFUN(I,K)+(FF(J,K+1)*C(I,J,K+1)-FF(J,K)*C(I,J,K))
       I=FLOAT(ICT)
       EPHI=1.0

C ***** INACCESSIBLE POPE VOLUMES TO SURFACTANT AND/OR POLYMER
   IF(I,EQ,3) EPHI=EPHI3
   IF(I,EQ,4) EPHI=EPHI4
   I=(K-1)*NCOMP+1
   OY(I)=(GFUN(I,K)-FFUN(I,K))/EPHI
36 CONTINUE
29 IADS=1
   RETURN
   END

```

```

      SUBROUTINE PROPRTY
C ***** THIS SUBPROGRAM CALCULATE NEW PHASE COMPOSITIONS AND PROPERTIES *****
C ***** DIMENSION ABPE(10),PHTOLD(10),F3OLD(10),IT(40) *****

C
   COMMON/NO/ICT,ICT1,ICT2,XICT,NCOMP
   COMMON/SOL/C(7,4,2),S(3,42),FF(3,42),NPHASE(42)
   COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEOP
   COMMON/CSEVIS/VIS1,VIS2,AP1,AP2,AP3,SSLOPE
   COMMON/SHEVIS/GAMHF,POHM,CSE1,RKMAX,RRK,ISHEAR
   COMMON/SYSTEM/UT,AMPERN,PHI,EPH13,EPH14,DISPJ(4)
   COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
   COMMON/ALPHA/ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5
   COMMON/PERMC/PERH(3,40),SRED(3,40),SN(3,40),VIS(3,40)
   COMMON/INJECT/DVP,VP,VT,VPI

C
   IPRNT=0

C ***** NEW PHASE COMPOSITIONS AND SATURATIONS *****
   CALL PHCOMP

C ***** CALCULATE PROPERTIES *****

C
   DO 28 K=1,ICT
     C3=C(3,4,K)+C(7,4,K)
     IF(NPHASE(K),EQ,1) GO TO 51
     DO 50 N=1,3
       IF(S(4,K),GT,0.991) GO TO 50
       WRITE(6,200)K
200  FORMAT(1X,'K=',13.3X,'NEGATIVE SATURATION APPEARED IN PROPRTY')
       RETURN
     51 CONTINUE
     51 CONTINUE

C ***** DETERMINE FOR WHICH PHASE VISCOSITY IS CALCULATED *****
   N1=N3=1
   N2=3
   IF(NPHASE(K),EQ,3)GO TO 65
   IF(NPHASE(K),EQ,1)GO TO 66
   IF(C3,LT,0.0001)GO TO 67
   IF(CSE(K),GE,CSEU,OR,NPHASE(K),EQ,4)N3=2
   IF(CSE(K),LE,CSEL,OR,NPHASE(K),EQ,5)N1=2
   GO TO 65
67 N2=2
   GO TO 65
66 IF(C3,LT,0.0001)N2=1
   IF(C3,GE,0.0001)N1=3
65 CONTINUE

C ***** PHASE NUMBER WHERE POLYMER EXISTS *****
   JP=1
   IF(N1,NE,1)JP=3

C ***** POLYMER EFFECT ON VISCOSITY EXCEPT SHEAR RATE EFFECT *****
   VISPP=VIS
   IF(C(1,4,K),LE,1.E-1)GO TO 60
C PERCENT REDUCTION FACTOR RK
   RK=1.0*(RKMAX-1.0)*N1+C(4,JP,K)/(1.0+RK+C(4,JP,K))
C SALINITY EFFECT
   CSE=CSE(K)

```

```

      IF(CSE(K) .LT. CSE1) CSE=CSE1
      VISPP=VIS1*RK*(1.+(AP1+C(4,JP,K)+AP2*C(4,JP,K)+2*AP3*C(4,JP,K)
      1**3)*CSE**SLOPE)
65 CONTINUE
C
C***** PHASE VISCOSITIES *****
      VIS(1,K)=VIS(2,K)=VIS(3,K)=1.0
      DO 69 N=1,N2,N3
      VISP=VIS1
      IF(N.EQ.JP) VISP=VISPP
      VIS(N,K)=C(1,N,K)*VISP*EXP(ALPHA1*(C(2,N,K)+C(3,N,K)))+
      1 C(2,N,K)*VIS2*EXP(ALPHA2*(C(1,N,K)+C(3,N,K)))+
      1 C(3,N,K)*ALPHA3*EXP(ALPHA4*(C(1,N,K)+ALPHA5*(C(2,N,K)
      IF(VIS(N,K).LT.VIS1) VIS(N,K)=VIS1
69 CONTINUE
      IF(NPHASE(K).EQ.1) GO TO 120
C
C***** TRAPPING FUNCTION AND RELATIVE PERMEABILITY *****
      IPHT=1
87 CONTINUE
      IPHT=IPHT+1
      PHTOLD(IPHT)=PHT(K)
      F3OLD(IPHT)=FF(3,K)
      IF(PHT(K).LT.1.E-4) PHT(K)=1.E-4
C
      CALL RELPERM(K)
      GO TO 19
C
C SINGLE PHASE FLOW
121 CONTINUE
      SRED(1,K)=SRED(2,K)=SRED(3,K)=0.0
      PERM(1,K)=PERM(2,K)=PERM(3,K)=0.0
      IF(C37.LT.0.0001) PERM(1,K)=1.0
      IF(C37.GE.0.0001) PERM(3,K)=1.0
C
C***** SHEAR RATE EFFECT ON POLYMER *****
19 CONTINUE
      IVIS=3
C
C CHECK WHETHER SHEAR RATE EFFECT IS NEEDED
      IF(IVIS.LE.1.AND.NPHASE(K).NE.1) GO TO 215
      IF(ISHEAR.EQ.1) GO TO 215
      IF(PERM(JP,K) .LE. 1.E-4) GO TO 215
      IF(S(JP,K) .LE. 1.E-4) GO TO 215
C
C SHEAR RATE DEPENDENT VISCOSITY
62 CONTINUE
      IVIS=IVIS+1
      REQ=(7.89536E-4*ABPERM/PH1)**0.5
      GAMMA=4.0*UT/REQ/PH1
      IF(NPHASE(K).NE.1) GAMMA=GAMMA*(FF(JP,K)**2/PERM(JP,K)/S(JP,K))
      1 **0.5
      IF(NPHASE(K).EQ.1) GAMMA=GAMMA*
      VISP=VIS1+(VISPP-VIS1)/(1.0+(GAMMA/GAMMAF)**(POW-1.0))
C
      VIS(JP,K)=C(1,JP,K)*VISP*EXP(ALPHA1*(C(2,JP,K)+C(3,JP,K)))+
      1 C(2,JP,K)*VIS2*EXP(ALPHA2*(C(1,JP,K)+C(3,JP,K)))+
      1 C(3,JP,K)*ALPHA3*EXP(ALPHA4*(C(1,JP,K)+ALPHA5*(C(2,JP,K)
      IF(VIS(JP,K) .LE. VIS1) VIS(JP,K)=VIS1
215 CONTINUE
C
C***** RELATIVE MOBILITY RATIO AND FRACTIONAL FLOW *****

```

```

      PH1=PERM(1,K)/VIS(1,K)
      PH2=PERM(2,K)/VIS(2,K)
      PH3=PERM(3,K)/VIS(3,K)
      PHT(K)=PH1+PH2+PH3
C
      FF(1,K)=PH1/PHT(K)
      FF(2,K)=PH2/PHT(K)
      FF(3,K)=1.0-FF(1,K)-FF(2,K)
C
C***** RECALCULATE VISCOSITY WITH SHEAR RATE EFFECT
      IF(ISHEAR.EQ.1) GO TO 70
      IF(C(4,0,K).LE.1.E-10) GO TO 70
      IF(PERM(JP,K).LE.1.E-8.OR.S(JP,K).LE.1.E-4) GO TO 70
      IF(IVIS.LE.1.AND.NPHASE(K).NE.1) GO TO 62
70 CONTINUE
C
C***** UPDATE PHT(K) AND RECALCULATE TRAPPING FUNCTION *****
      IF(NPHASE(K).EQ.1) GO TO 204
      ABPE(IPHT)=PHT(K)-PHTOLD(IPHT)
      RELPE=ABPE/ABPE(IPHT)/PHTOLD(IPHT)
      IF(RELPE.LT.0.001) GO TO 204
      IF(IPHT.GE.10) GO TO 205
      IF(IPHT.GE.2) GO TO 84
      GO TO 40
84 CONTINUE
      PHT(K)=PHTOLD(IPHT)-ABPE(IPHT)*(PHTOLD(IPHT)-PHTOLD(IPHT-1))/
      1 (ABPE(IPHT)-ABPE(IPHT-1))
      GO TO 40
C
C UNLESS CONVERGED, PRINT MESSAGE AND SHOW ITERATION
205 WRITE(6,990) VP,K
990 FORMAT('///1X,*PHT DID NOT CONVERGE*/1X,*VP=*,F6.4/1X,* K=*,I2//
      1 15X,2HP,4X,6HPHTOLD,9X,4HARPE,12X,5HF3OLD)
991 DO 994 IP=1,IPHT
991 WRITE(6,994) IP,PHTOLD(IP),ARPE(IP),F3OLD(IP)
994 FORMAT(15X,I2,3E15.7)
      STOP
204 CONTINUE
      IF(NPHASE(K).EQ.1) IPHT=1
      IT(K)=IPHT
      F3OLD(IPHT+1)=FF(3,K)
C
C***** MAKE ALL NON-EXISTING VISCOSITY EQUAL ZERO *****
      IF(NPHASE(K).EQ.3) GO TO 75
      IF(NPHASE(K).EQ.1) GO TO 76
      IF(C37.LT.0.0001) GO TO 77
      IF(CSE(K).GE.CSEL.OR.NPHASE(K).EQ.4) VIS(2,K)=0.0
      IF(CSE(K).LE.CSEL.OR.NPHASE(K).EQ.5) VIS(1,K)=0.0
      GO TO 75
77 VIS(3,K)=0.0
      GO TO 75
75 IF(C37.LT.0.0001) VIS(2,K)=VIS(3,K)=0.0
      IF(C37.GE.0.0001) VIS(1,K)=VIS(2,K)=0.0
75 CONTINUE
20 CONTINUE
      IF(IPHT.NE.1) GO TO 404
C
C***** PRINT NO. OF ITERATION AND CONVERGENCE AT EACH BLOCK *****
      PRINT 981,VP,K,N=1,ICT)
      PRINT 982,(IT(K),K=1,ICT)
      PRINT 783,NPHASE(K)
      IF(NPHASE(K).EQ.1) GO TO 971

```



```

DO 974 IP=1,IPHT
974 PRINT 980,IP,PHTOLD(IP),ABPE(IP),F3OLD(IP+1)
971 CONTINUE
981 FORMAT(//5X,*VP **F7.0//8X,*BLOCK **,40I3)
982 FORMAT(8X,* IT **,40I3)
983 FORMAT(/15X,*AT 40 TH BLOCK (NPHASE **,13,*)*)
1 20X,2MIP,4X,3MPT,12X,4HABPE,11X,3HFF3)
984 FORMAT(20X,12,3F15.7)
980 CONTINUE
RETURN
END

```

```

SUBROUTINE RELPER4(K)
C-----
C THIS SUBPROGRAM GIVES RESIDUAL SATURATION AND RELATIVE PERMEABILITY.
C WHEN THREE PHASES APPEAR, A RELATIVE PERMEABILITY MODEL IS SELECTED
C ACCORDING TO IPERM.
C IPERM = 0 : POPE'S MODEL
C = 1 : HIRASAKI'S MODEL
C = 2 : MODIFIED HIRASAKI'S MODEL
C = 3 : LAKE'S MODEL
C-----
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEOP
COMMON/PERM/IPERM,P1RW,P2RW,E1,E2,P1RC,P2RC
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
COMMON/PERF-C/PERH(3,40),SRED(3,40),SN(3,40),VIS(3,40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFT4
COMMON/RESTD/S1R,S2R,S3R
C
C CALCULATE RESIDUAL SATURATIONS BASED ON CAPILLARY NUMBER
S1R=S1RW*(1.0+T11*(ALOG10(PHT(K))+XIFT1(K)+T12))
S2R=S2RW*(1.0+T21*(ALOG10(PHT(K))+XIFT2(K)+T22))
IF(S1R.LE.0.0)S1R=0.0
IF(S2R.LE.0.0)S2R=0.0
IF(S1R.GT.S1RW)S1R=S1RW
IF(S2R.GT.S2RW)S2R=S2RW
IF(NPHASE(K).EQ.3)GO TO 10
C
C***** TWO PHASE FLOW *****
C
C DEFINE WETTING PHASE AND NON-WETTING PHASE SATURATIONS
SMET=S(1,K)
SNOW=S(2,K)
C37=C(3,4,K)+C(7,4,K)
IF(C37.LT.0.0001)GO TO 201
IF(CSE(K).GE.CSEU,OR,NPHASE(K).EQ.0)SNOW=S(3,K)
IF(CSE(K).LE.CSEL,OR,NPHASE(K).EQ.5)SMET=S(3,K)
201 CONTINUE
C
C DETERMINE RESIDUAL SATURATION
IF(SMET.LT.S1R)S1R=SMET
IF(SNOW.LT.S2R)S2R=SNOW
S3R=0.0
C
C NORMALIZED SATURATION
SN(1,K)=(SMET-S1R)/(1.0-S1R-S2R)
IF(SN(1,K).LT.0.0)SN(1,K)=0.0
IF(SN(1,K).GT.1.0)SN(1,K)=1.0
SN(2,K)=1.0-SN(1,K)
SN(3,K)=0.0
C
C END POINT AND CURVATURE OF RELATIVE PERMEABILITY CURVE
P1R=P1RW*(S2RW-S2R)*(P1RC-P1RW)/S2RW
P2R=P2RW*(S1RW-S1R)*(P2RC-P2RW)/S1RW
E1C=1.0*(E1-1.0)*S2R/S2RW
E2C=1.0*(E2-1.0)*S1R/S1RW
C
C RELATIVE PERMEABILITY, RESIDUAL SATURATION, NORMALIZED SATURATION
PERM(1,K)=P1R*40S(S1(1,K))+E1C
PERM(2,K)=P2R*40S(SN(2,K))+E2C
PERM(3,K)=1.0

```

```

IF(C17,LT,0,NOR1)GO TO 99
IF(CSE(K),GE,CSEU,OR,NPHASE(K),EQ,4)GO TO 20
PERM(3,K)=PERM(1,K)
PERM(1,K)=0.0
S1P=S1P
S1R=0.0
SN(3,K)=SN(1,K)
SN(1,K)=0.0
GO TO 99
20 CONTINUE
PERM(3,K)=PERM(2,K)
PERM(2,K)=0.0
S3P=S2P
S2R=0.0
SH(3,K)=SN(2,K)
SN(2,K)=0.0
GO TO 99
C
C***** THREE PHASE FLOW *****
10 CONTINUE
IF(IPERM,EQ,0)CALL POPE(K)
IF(IPERM,EQ,1)CALL HIRA(K)
IF(IPERM,EQ,2)CALL OMNO(K)
IF(IPERM,EQ,3)CALL LAKE(K)
C
99 CONTINUE
C
C RENAME RESIDUAL SATURATIONS
SRED(1,K)=S1R
SRED(2,K)=S2R
SRED(3,K)=S3R
RETURN
END

```

```

SUBROUTINE POPE(K)
C-----
C THIS IS A THREE PHASE RELATIVE PERMEABILITY MODEL,
C THIS MODEL WAS USED BY R.A. POPE IN HIS ORIGINAL SIMULATOR.
C-----
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/PERM/IPERM,N1R4,P2RW,E1,E2,P1RC,P2RC
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
COMMON/PERMC/PERN(3,40),SRED(3,40),SN(3,40),VIS(3,40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTW
COMMON/RESID/S1R,S2R,S3R
C
P3R=E3*1.0
S3R=T31+T32*(ALOG10(PHT(K))+XIFT3(K))
IF(S(1,K),LT,S1R)S1R=S(1,K)
IF(S(2,K),LT,S2R)S2R=S(2,K)
SM=1.0-(S1P+S2P+S3P)
SN(1,K)=(S(1,K)-S1R)/SM
SN(2,K)=(S(2,K)-S2R)/SM
IF(SN(1,K),LT,0.0)SN(1,K)=0.0
IF(SN(2,K),LT,0.0)SN(2,K)=0.0
IF(SN(1,K),GT,1.0)SN(1,K)=1.0
IF(SN(2,K),GT,1.0)SN(2,K)=1.0
SN(3,K)=1.0-(SN(1,K)+SN(2,K))
P1R=P1RW+(S2RW-S2R)*(P1RC-P1RW)/S2RW
P2R=P2RW+(S1RW-S1R)*(P2RC-P2RW)/S1RW
C
PERN(1,K)=P1R*SN(1,K)
PERN(2,K)=P2R*SN(2,K)
PERN(3,K)=P3R*SN(3,K)*E3
RETURN
END

```

SUBROUTINE HIRA(K)

```

C-----
C THIS IS A THREE PHASE RELATIVE PERMEABILITY MODEL.
C THIS MODEL WAS PRESENTED BY G. HIRASAKI.
C-----
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/PERM/IPERM,P1RW,P2RW,E1,E2,P1RC,P2RC
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
COMMON/PERMC/PERM(3,40),SRFD(3,40),SN(3,40),VIS(3,40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTW
COMMON/RESID/S1R,S2R,S3R

```

```

C
C RESIDUAL SATURATION
S32R=S2RW*(1.+T21*(ALOG10(PHT(K))+XIFT1(K)+T22))
S31R=S1RW*(1.+T11*(ALOG10(PHT(K))+XIFT2(K)+T12))
IF(S32R.LT.0.)S32R=0.
IF(S31R.LT.0.)S31R=0.
IF(S32R.GT.S2RW)S32R=S2RW
IF(S31R.GT.S1RW)S31R=S1RW
S1RP=S31R-S(3,K)
S2RP=S32R-S(3,K)
S3R1=S32R-S(2,K)
S3R2=S31R-S(1,K)
S1R=AMAX1(S1R,S1RP)
S2R=AMAX1(S2R,S2RP)
S3R=AMAX1(S3R1,0.,S3R2)

```

```

C
C NORMALIZED SATURATION
SN(1,K)=(S(1,K)-S1R)/(1.-(S32R+S1R))
SN(2,K)=(S(2,K)-S2R)/(1.-(S31R+S2R))
SN(3,K)=(S(3,K)-S3R)/(1.-(S1R+S2R+S3R))
IF(SN(1,K).LT.0.)SN(1,K)=0.
IF(SN(2,K).LT.0.)SN(2,K)=0.
IF(SN(3,K).LT.0.)SN(3,K)=0.

```

```

C
C END POINT AND CURVATURE OF RELATIVE PERMEABILITY CURVE
S1T=AMIN1(S1R,S(1,K))
S2T=AMIN1(S2R,S(2,K))
WE1=S2T/(S1T+S2T)
P1R=P1RC-(P1RC-P1RW)*S32R/S2RW
P2R=P2RC-(P2RC-P2RW)*S31R/S1RW
P3R=WE1*P1R+(1.-WE1)*P2R
E1C=1.+(E1-1.)*S32R/S2RW
E2C=1.+(E2-1.)*S31R/S1RW
E3C=WE1*E1C+(1.-WE1)*E2C

```

```

C
C RELATIVE PERMEABILITY
PERM(1,K)=P1R*SN(1,K)**E1C
PERM(2,K)=P2R*SN(2,K)**E2C
PERM(3,K)=P3R*SN(3,K)**E3C
RETURN
END

```

SUBROUTINE OMNO(K)

```

C-----
C THIS IS A THREE PHASE RELATIVE PERMEABILITY MODEL.
C THIS IS A MODIFIED VERSION OF HIRASAKI'S MODEL.
C THIS SUBPROGRAM IS NOT COMPLETED YET (NEED MORE MODIFICATION).
C-----

```

```

COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/PERM/IPERM,P1RW,P2RW,E1,E2,P1RC,P2RC
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
COMMON/PERMC/PERM(3,40),SRFD(3,40),SN(3,40),VIS(3,40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTW
COMMON/RESID/S1R,S2R,S3R

```

```

C
C RESIDUAL SATURATION
S32R=S2RW*(1.+T21*(ALOG10(PHT(K))+XIFT1(K)+T22))
S31R=S1RW*(1.+T11*(ALOG10(PHT(K))+XIFT2(K)+T12))
IF(S32R.LT.0.)S32R=0.
IF(S31R.LT.0.)S31R=0.
IF(S32R.GT.S2RW)S32R=S2RW
IF(S31R.GT.S1RW)S31R=S1RW
S1R=AMIN1(S(1,K),S1R)
S2R=AMIN1(S(2,K),S2R)
S3R1=S31R-S1R
S3R2=S32R-S2R
S3R=AMAX1(S3R1,0.,S3R2)
IF(S(3,K).LT.S3R)S3R=S(3,K)

```

```

C
C END POINT AND CURVATURE OF RELATIVE PERMEABILITY CURVE
IF((S1R+S2R).LT.1.E-4)GO TO 10
S32RC=AMIN1(S32R,(S3R+S2R))
S31RC=AMIN1(S31R,(S3R+S1R))
WE1=S2R/(S1R+S2R)
P1R=P1RC-(P1RC-P1RW)*S32RC/S2RW
P2R=P2RC-(P2RC-P2RW)*S31RC/S1RW
P3R=WE1*P1R+(1.-WE1)*P2R
E1C=1.+(E1-1.)*S32RC/S2RW
E2C=1.+(E2-1.)*S31RC/S1RW
E3C=WE1*E1C+(1.-WE1)*E2C

```

```

C
C NORMALIZED SATURATION
SMUR=1.-(S1R+S2R+S3R)
SN(1,K)=(S(1,K)-S1R)/SMUR
SN(2,K)=(S(2,K)-S2R)/SMUR
SN(3,K)=(S(3,K)-S3R)/SMUR
IF(SN(1,K).LT.0.)SN(1,K)=0.
IF(SN(2,K).LT.0.)SN(2,K)=0.
IF(SN(3,K).LT.0.)SN(3,K)=0.

```

```

C
C RELATIVE PERMEABILITY
PERM(1,K)=P1R*SN(1,K)**E1C
PERM(2,K)=P2R*SN(2,K)**E2C
PERM(3,K)=P3R*SN(3,K)**E3C
RETURN

```

```

10 DO 20 J=1,3
20 PERM(J,K)=S(J,K)
RETURN
END

```

```

SUBROUTINE LAKF(K)
C-----
C THIS IS A THREE PHASE RELATIVE PERMEABILITY MODEL.
C THIS MODEL WAS DEVELOPPED BY L. LAKE.
C-----
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/PERM/PERM,P1RW,P2RW,E1,E2,P1RC,P2RC
COMMON/TRAP/T11,T12,T21,T22,T31,T32,S1RW,S2RW,PHT(40)
COMMON/PERMC/PERMC(3,40),SRED(3,40),SN(3,40),VIS(3,40)
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTW
COMMON/RESID/S1R,S2R,S3R

C
GLAKE=S(2,K)*(1-S(1,K))/(S(1,K)+S(2,K))
S3R=S2R+GLAKE*(S1R-S2R)

C
C ACTUAL RESIDUAL SATURATION
S1R=AMIN1(S1R,S(1,K))
S2R=AMIN1(S2R,S(2,K))
S3R=AMIN1(S3R,S(3,K))

C
C INTERPOLATION FACTORS
S1IP=(S2RW-S2R)/S2RW
S2IP=(S1RW-S1R)/S1RW

C
C END POINTS OF RELATIVE PERMEABILITIES
P1R=P1RW+S1IP*(P1RC-P1RW)
P2R=P2RW+S2IP*(P2RC-P2RW)
P3R=P2R+GLAKE*(P1R-P2R)

C
C CURVATURE OF RELATIVE PERMEABILITY CURVES
E1C=E1+S1IP*(1-E1)
E2C=E2+S2IP*(1-E2)
E3C=E2C+GLAKE*(E1C-E2C)

C
C NORMALIZED SATURATION
SMOR=1-(S1R+S2R+S3R)
SN(1,K)=(S(1,K)-S1R)/SMOR
SN(2,K)=(S(2,K)-S2R)/SMOR
SN(3,K)=(S(3,K)-S3R)/SMOR

C
C RELATIVE PERMEABILITIES
PERM(1,K)=P1R*SN(1,K)**E1C
PERM(2,K)=P2R*SN(2,K)**E2C
PERM(3,K)=P3R*SN(3,K)**E3C

C
RETURN
END

```

```

SUBROUTINE PHCOMP
C-----
C THIS SUBPROGRAM GIVES PHASE COMPOSITIONS, SATURATIONS, AND IFT.
C MODIFIED HANCO EQUATIONS ARE USED TO OBTAIN PHASE EQUILIBRIUM.
C MOST SURFACTANT RICH PHASE IS DEFINED AS PHASE 3.
C-----
COMMON/MD/ICT1,ICT2,XICT,MCOMP
COMMON/IFT/G11,G12,G13,G21,G22,G23
COMMON/A/A11,A12,A21,A22
COMMON/PHASE/F1,F2,F3,R1,R2,R3,C2PLC,C2PRC
COMMON/CSE/CSE(42),CSEL,CSEU,RCSE,CSEOP
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/TRY/A,B,G,F,IFX,ALPHA,BETA,KK
COMMON/XIFT/XIFT1(42),XIFT2(42),XIFT3(42),XIFTW

C
DO 101 K=1,ICT2
KK=K
IF(K.EQ. ICT1) GO TO 101

C
C COMBINE SURFACTANT AND ALCOHOL TOGETHER AS COMPONENT THREE
IF(C(7,4,K).GE. 1.0E-4) RSAC=C(3,4,K)/C(7,4,K)
C(3,4,K)=C(3,4,K)+C(7,4,K)

C
C TEST FOR CHEMICAL
IF(C(3,4,K).LT.0.0001) GO TO 60
IF(CRE(K).LT.CSEU) GO TO 20

C
C***** TYPE II(+) BEHAVIOR : HIGH SALINITY *****
A1=A11+A12+CSE(K)
IF(C(2,4,K).LT.0.0001) GO TO 25
R32=C(3,4,K)/C(2,4,K)
R31=(R32/A1)**(1.0/B1)
C(1,3,K)=R32/(R32+R32*R31+R31)
C(2,3,K)=1.0-(R31+1.0)*C(1,3,K)
C(3,3,K)=1.0-C(1,3,K)-C(2,3,K)
IF(C(3,3,K).LT.C(3,4,K)) GO TO 25
NPHASE(K)=2
IF(ABS(C2PLC).GT..00001.AND.ABS(F1).GT..00001)GO TO 12
C(1,1,K)=1.0
C(2,1,K)=0.0
C(3,1,K)=0.0
GO TO 11

12
A=A1
R=R1
F=F1
C3PLC=S*(-A*C2PLC+((A*C2PLC)**2+4.*A*C2PLC*(1.-C2PLC))**.5)
R=(1.-C3PLC-C2PLC)/C2PLC
C
C CALCULATE R31 ,R32 AT PLAIT POINT
XR=C3PLC/C2PLC-1.E-16
XL=1.0F-7

C
C CALCULATE THE CONCENTRATIONS OF COMPONENTS OF TWO PHASES
CALL TIELINE(XR,XL)
DO 13 J=1,6
13 C(1,J,K)=C(1,2,K)
11 DO 15 J=1,4
15 C(1,2,K)=0.0
DO 10 J=5,6
DO 14 J=1,3,2
14 C(1,J,K)=C(1,4,K)*C(1,J,K)/C(1,4,K)
S(1,K)=(C(1,4,K)-C(1,3,K))/(C(1,1,K)-C(1,3,K))
S(3,K)=1.0-S(1,K)

```

```

      S(2,K)=0.0
      C(4,1,K)=C(4,0,K)/S(1,K)
      C(4,3,K)=0.0
C CALCULATE INTERFACIAL TENSION BETWEEN MICROEMULSION AND WATER
      XIFT1(K)=G12+G11/(C13+C(1,3,K)/C(3,3,K)+1.0)
      XIFT2(K)=XIFT1(K)
      GO TO 100
20 C IF(CSE(K),GT,CSEL) GO TO 30
C ***** TYPE II(-) BEHAVIOR : LOW SALINITY *****
      A2=A21+A22+CSE(K)
      R31=C(3,4,K)/C(1,4,K)
      R32=A2+R31+R2
      C(1,3,K)=R32/(R32+R32+R31+P31)
      C(2,3,K)=1.0-(R31+1.0)*C(1,3,K)
      C(3,3,K)=1.0-C(1,3,K)-C(2,3,K)
      IF(C(3,3,K).LT,C(3,4,K)) GO TO 25
      NPHASE(K)=2
      IF(ABS(C2PRC).LT,1.0,AND,ABS(F2).GT,.000001)GO TO 19
      C(1,2,K)=0.0
      C(2,2,K)=1.0
      C(3,2,K)=0.0
      GO TO 18
19 A=A21+A22+CSE(K)
      H=R2
      F=F2
      C3PRC=.5*(-A+C2PPC+((A+C2PRC)**2+.4,.4A+C2PRC*(1.-C2PRC))**.5)
      G=(1.-C3PRC-C2PRC)/C2PRC
      XR=G+R1+.4F
      XL=1.0E-7
C CALCULATE THE CONCENTRATIONS OF COMPONENTS OF TWO PHASES
      CALL TIELINE(XR,XL)
      DO 21 I=1,6
23 C(1,3,K)=C(1,1,K)
10 DO 22 I=1,6
22 C(1,1,K)=H.0
      DO 21 I=5,6
      DO 21 J=2,3
21 C(1,J,K)=C(1,4,K)*C(1,J,K)/C(1,4,K)
      S(3,K)=(C(1,4,K)-C(1,2,K))/(C(1,3,K)-C(1,2,K))
      S(2,K)=1.0-S(3,K)
      S(1,K)=0.0
      C(4,3,K)=C(4,4,K)/S(3,K)
      C(4,2,K)=0.0
C CALCULATE INTERFACIAL TENSION BETWEEN MICROEMULSION AND OIL
      XIFT2(K)=G22+G21/(R23+C(2,3,K)/C(3,3,K)+1.0)
      XIFT1(K)=XIFT2(K)
      GO TO 100
C
C ***** TYPE III BEHAVIOR : INTERMEDIATE SALINITY *****
30 C2M=(CSE(K)-CSEI)/(CSEU-CSEL)
      IF(CSE(K)=.5*(CSEU+CSEL))301,301,302
301 A3=A21+A22+CSE(K)
      GO TO 303
302 A3=A11+A12+CSE(K)
303 A=A3
      H=R3
      F=F3
      C3M=.5*(-A+C2M+((A+C2M)**2+.4,.4A+C2M*(1.-C2M))**.5)
      C1M=1.-C2M-C3M
      IF(C2M.LT,1.0E-10) C2M=1.0E-10
      IF(C1M.LT,1.0E-10) C1M=1.0E-10

```

```

      IF(C(2,4,K),GT,C2M)GO TO 40
C TEST FOR THREE PHASES
      IF(C(3,4,K).LT,C3M+C(2,4,K)/C2M) GO TO 50
C
C TYPE II(+) LURE
      IF(C(2,4,K).LE,0.001) GO TO 25
      NPHASE(K)=4
      R32=C(3,4,K)/C(2,4,K)
      R31=(R32/A3)**(1.0/R3)
      C(1,3,K)=R32/(R32+R32+R31+R31)
      C(2,3,K)=1.0-(R31+1.0)*C(1,3,K)
      C(3,3,K)=1.0-C(1,3,K)-C(2,3,K)
      IF(C(3,3,K).LT,C(3,4,K))GO TO 25
      IF(ABS(C2PLC).GT,1.0E-10)GO TO 304
      C(1,1,K)=1.0
      C(2,1,K)=0.0
      C(3,1,K)=0.0
      GO TO 314
304 C2PL=C2PLC+(CSE(K)-CSEU)*C2PLC/(CSEU-CSEL)
      C3PL=.5*(-A+C2PL+((A+C2PL)**2+.4,.4A+C2PL*(1.-C2PL))**.5)
      C1PL=1.-C2PL-C3PL
      ALPHA=C3M/C2M
      BETA=SQRT(C2M**2+C3M**2)/C2M
      G=(1.+(ALPHA-BETA)*C2PL-C3PL)/(BETA+C2PL)
      XR=C3PL/C1PL-1.0E-16
      XL=1.0E-7
      CALL TIELINE(XR,XL)
      DO 31 I=1,6
314 C(1,2,K)=0.0
33 IF(C(3,4,K),GT,C(3,3,K)) GO TO 25
      DO 31 I=5,6
      DO 31 J=1,3,2
31 C(1,J,K)=C(1,4,K)*C(1,J,K)/C(1,4,K)
      S(1,K)=(C(1,4,K)-C(1,3,K))/(C(1,1,K)-C(1,3,K))
      S(3,K)=1.0-S(1,K)
      S(2,K)=0.0
      C(4,1,K)=C(4,4,K)/S(1,K)
      C(4,3,K)=0.0
      XIFT1(K)=G12+G11/(R13+C(1,3,K)/C(3,3,K)+1.0)
      XIFT2(K)=XIFT1(K)
      GO TO 100
40 IF(C(3,4,K).LT,C3M+C(1,4,K)/C1M) GO TO 50
C
C TYPE II(-) LURE
      NPHASE(K)=5
      R31=C(3,4,K)/C(1,4,K)
      R32=A3+R31+R3
      C(1,3,K)=R32/(R32+R32+R31+R31)
      C(2,3,K)=1.0-(R31+1.0)*C(1,3,K)
      C(3,3,K)=1.0-C(1,3,K)-C(2,3,K)
      IF(C(3,4,K),GT,C(3,3,K)) GO TO 25
      IF(ABS(C2PRC).LT,1.0)GO TO 305
      C(1,2,K)=0.0
      C(2,2,K)=1.0
      C(3,2,K)=0.0
      GO TO 315
305 C2PR=C2PRC*(1.-C2PRC)+(CSE(K)-CSEL)/(CSEU-CSEL)
      C3PR=.5*(-A+C2PR+((A+C2PR)**2+.4,.4A+C2PR*(1.-C2PR))**.5)
      C1PR=1.-C2PR-C3PR
      ALPHA=C3M/C1M
      BETA=SQRT(C3M**2+C1M**2)/C1M
      G=BETA*C1PR/(1.+(ALPHA-BETA)*C1PR-C3PR)

```

```

      XH=G*R31+*F-1,F=16
      XL=1,0F=7
      CALL TIE(LINE(XR,XL)
315 DO 43 I=1,6
43  C(1,1,K)=0,0
      DO 41 J=5,6
      DO 41 J=2,3
41  C(1,J,K)=C(1,4,K)+C(1,J,K)/C(1,4,K)
      S(2,K)=(C(2,4,K)-C(2,3,K))/(C(2,2,K)-C(2,3,K))
      S(3,K)=1.-S(2,K)
      S(1,K)=0,0
      C(4,1,K)=C(4,4,K)/S(3,K)
      C(4,2,K)=0,0
      XIFT2(K)=G22+G21/(G23+C(2,3,K)/C(3,3,K)+1,0)
      XIFT1(K)=XIFT2(K)
      GO TO 100

C
C THREE PHASES
50  NPHASE(K)=3
      C(1,1,K)=1,0
      C(2,1,K)=0,0
      C(3,1,K)=0,0
      C(1,2,K)=0,0
      C(1,2,K)=0,0
      C(2,2,K)=1,0
      C(3,2,K)=0,0
      C(1,3,K)=C1H
      C(2,3,K)=C2H
      C(3,3,K)=C3H
      DO 51 J=1,3
      DO 51 I=5,6
51  C(1,J,K)=C(1,4,K)+C(1,J,K)/C(1,4,K)
      S(2,K)=(C(1,4,K)-C(1,3,K))+C(2,3,K)+(1,0-C(1,3,K))*(C(2,4,K)
      1-C(2,3,K))/(1,0-C(1,3,K)-C(2,3,K))
      S(1,K)=(C(1,4,K)-C(1,3,K))+C(1,0-C(2,3,K))+C(2,4,K)-C(2,3,K))
      1+C(1,3,K))/(1,0-C(1,3,K)-C(2,3,K))
      S(3,K)=1.-S(1,K)-S(2,K)
      C(4,1,K)=C(4,4,K)/S(1,K)
      C(4,2,K)=0,0
      C(4,3,K)=0,0

C CALCULATE TWO INTERFACIAL TENSIONS
      XIFT1(K)=G12+G11/(G13+C(1,3,K)/C(3,3,K)+1,0)
      XIFT2(K)=G22+G21/(G23+C(2,3,K)/C(3,3,K)+1,0)
      XIFT3(K)=AMIN1(XIFT1(K),XIFT2(K))
      GO TO 100

C
C***** SINGLE PHASE REGION : TYPE II(-), TYPE II(+), TYPE I *****
25  NPHASE(K)=1
      DO 26 I=1,6
      C(1,1,K)=C(1,2,K)=0,0
      C(1,3,K)=C(1,4,K)
26  CONTINUE
      S(1,K)=0,0
      S(2,K)=0,0
      S(3,K)=1,0
      GO TO 100

C
C***** NO CHEMICAL *****
40  S(2,K)=C(2,4,K)
      S(1,K)=1.-S(2,K)
      S(3,K)=0,0
      IF(S(1,K),LT,1,E-12)GO TO 97
      C(3,1,K)=C(3,4,K)/(C(1,4,K)+C(3,4,K))

```

```

      C(1,1,K)=1,0-C(3,1,K)
97  CONTINUE
      DO 98 J=2,3
      DO 98 I=1,6
98  C(1,J,K)=0,0
      C(2,2,K)=1,0
      C(2,1,K)=0,0
      DO 70 J=1,3
      DO 70 I=1,NCOMP
70  C(1,J,K)=0,0
      IF(S(1,K),LT,1,E-12)GO TO 72
      C(3,1,K)=C(3,4,K)/S(1,K)
      C(1,1,K)=1,0-C(3,1,K)
      C(4,1,K)=C(4,4,K)/S(1,K)
      C(5,1,K)=C(5,4,K)/S(1,K)
      C(6,1,K)=C(6,4,K)/S(1,K)
      IF(S(2,K),LT,1,E-12)GO TO 74
      C(2,2,K)=1,0
      NPHASE(K)=2
      XIFT1(K)=XIFT2(K)=XIFTW
      GO TO 100
72  C(2,2,K)=1,0
74  NPHASE(K)=1

C
C
C SEPARATE SURFACTANT AND ALCOHOL FROM COMPONENT THREE
100  IF(C(7,4,K).GE.1,0E-8)GO TO 102
      DO 103 J=1,4
103  C(7,J,K)=0,0
      GO TO 101
102  DO 104 J=1,4
      C(7,J,K)=C(3,J,K)/(1,0+RSA)
      C(3,J,K)=C(7,J,K)*RSA
104  CONTINUE
101  CONTINUE
      RETURN
      END

```

```

SUBROUTINE TIELINE(XR,XL)
-----
C THIS IS A ROOT-FINDING PROGRAM WITH BISECTION METHOD.
C CONVERGENCE IS CHECKED WITH THE VALUE OF FUNCTION FXAPP.
C TOLERANCE EPSTIE IS FIXED IN THIS SUBPROGRAM.
C EPSTIE MAY NEED BE CHANGED.
-----
      EPSTIE=.001
      IFX=.0
      CALL TRY(XR,FXR)
      CALL TRY(XL,FXL)
2     APP=(XL+XR)/2.
      IFX=IFX+1
      IF(IFX,LE,50)GO TO 3
      PRINT 100
100  FORMAT('10X,APHASE COMPOSITION DID NOT CONVERGE IN TIELINE*')
      STOP
3     CONTINUE
      CALL TRY(APP,FXAPP)
      IF(FXAPP*FXL)7,9,8
7     XR=APP
      FXR=FXAPP
      IF(ABS(FXAPP).LE,EPSTIE) GO TO 9
      GO TO 2
8     XL=APP
      FXL=FXAPP
      IF(ABS(FXAPP).LE,EPSTIE) GO TO 9
      GO TO 2
9     CONTINUE
      RETURN
      END

```

```

SUBROUTINE TRY(X,FX)
-----
C THIS SUBPROGRAM GIVES PHASE COMPOSITIONS.
C THIS PROGRAM IS USED WHEN THE PLAI POINT IS NOT AT CORNER.
C EQUATIONS OF BINODAL CURVE AND DISTRIBUTION CURVE ARE USED.
-----
      COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
      COMMON/TRY/A,B,G,F,IFX,ALPHA,BETA,KK
C
      KK=KK
      IF(NPHASE(KK)=4)10,20,30
10     Y=(X/A)**(1./B)
      Z=G**(-1./F)*X**(-1./F)
      IF(Z,GT,1.E-10)GO TO 12
      PRINT 100
100  FORMAT('10X,*Z IS LESS THAN 1.E-10 IN SUBROUTINE TRY*')
      STOP
12     CONTINUE
      W=A*Z**B
      C(1,2,K)=X/(X+Y+Z)
      C(2,2,K)=1.-(1.+Y)*C(1,2,K)
      C(3,2,K)=1.-C(1,2,K)-C(2,2,K)
      C(1,1,K)=W/(Z+Z*W+W)
      C(2,1,K)=1.-(1.+Z)*C(1,1,K)
      C(3,1,K)=1.-C(1,1,K)-C(2,1,K)
      FX=(C(3,2,K)-C(3,4,K))*C(2,1,K)-C(2,4,K)-(C(3,1,K)-C(3,4,K))*
      1(C(2,2,K)-C(2,4,K))
      RETURN
CC     Y=R32(1) X=R31(1) W=R32(3) Z=R31(3)
CC     LEFT NODE
C
20     Y=A*X**B
      C(2,1,K)=X/(X+Y+Z)
      C(3,1,K)=C(2,1,K)*Y
      C(1,1,K)=1.-C(2,1,K)-C(3,1,K)
      P=G*((C(3,1,K)-C(2,1,K)*ALPHA)/(1.-C(3,1,K)+C(2,1,K)*(ALPHA-
      1BETA)))*F
      W=BETA*(P+ALPHA/BETA)
      Z=(W/A)**(1./B)
      C(2,3,K)=Z/(Z+Z*W+W)
      C(3,3,K)=C(2,3,K)*W
      C(1,3,K)=1.-C(2,3,K)-C(3,3,K)
      FX=(C(3,3,K)-C(3,4,K))*C(2,1,K)-C(2,4,K)-(C(3,1,K)-C(3,4,K))*
      1(C(2,3,K)-C(2,4,K))
      RETURN
C
C     CALCULATE THE CONC. OF WATER, OIL, SURFACTANT IN EACH PHASE
C     AT RIGHT NODE (TYPE TII) PHASE BEHAVIOR.
C     X=R32(2) Y=R31(2) W=R32(3) Z=R31(3)
C
30     Y=(X/A)**(1./B)
      C(2,2,K)=Y/(X+Y+Z)
      C(3,2,K)=C(2,2,K)*X
      C(1,2,K)=1.-C(2,2,K)-C(3,2,K)
      P=((C(3,2,K)-ALPHA*C(1,2,K))/G/(1.+C(1,2,K)*(ALPHA-BETA)-
      1(C(3,2,K))**(-1./F)
      Z=BETA*(P+ALPHA/BETA)
      W=A*Z**B
      C(2,3,K)=Z/(Z+Z*W+W)
      C(3,3,K)=C(2,3,K)*W

```

```

C(1,3,K)=1.-C(2,3,K)-C(3,3,K)
FX=(C(3,2,K)-C(3,4,K))*(C(2,3,K)-C(2,4,K))-(C(3,3,K)-C(3
1,4,K))*(C(2,2,K)-C(2,4,K))
RETURN
END

```

SUBROUTINE IONCNG

```

C-----
C  THIS SUBPROGRAM GIVES CATION EXCHANGE BETWEEN CLAY AND MORIL PHASES
C-----
C
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/ADSORP/CAOSS(44),C4AOSS(44),C6AOSS(44),C6HATS(44)
COMMON/CA/CBAOSS(44),CC6(42)
COMMON/ION/FFOV,DC3,K
COMMON/HHR/XKC,XK94,XK96,XKHAT,OV
COMMON/HI9/C3R,C6R,C6HAT,9R3,96,FT3,FT6,S1K,C8,C8R
ITER=0
C  ALL CONCENTRATIONS MUST BE IN UNITS OF MEQ/ML
C3R=C3AOSS(K)
J=1
IF(S(1,K),LE,0,0)J=3
IF(NPHASE(K),EQ,1)J=4
C(6,J,K)=C(1,J,K)*C(6,4,K)/C(1,4,K)
C(3,J,K)=C(1,J,K)*C(3,4,K)/C(1,4,K)
IF(XKC,LE,1,9E-10)GO TO 48
IF(C(3,1,K),LE,1,9E-5)GO TO 48
C8R=CBAOSS(K)
CC6(K)=CC6(K)*S(1,K)+FFOV*(CC6(K+1)+FF(1,K+1)-
1CC6(K)*FF(1,K))
9R3=C(3,4,K)+C3R+C8R+CC6(K)
96=C(6,4,K)+C6AOSS(K)+2.*CC6(K)+C6HATS(K)+2.*CBAOSS(K)
C
C  ION EXCHANGE WITH SURFACTANT COMPLEX
C  USING NEWTON RAPHSON ITERATION METHOD TO FIND THE CORRET
C  VALUES OF C31 AND C61
ITER=0
C5=C(5,1,K)
C3=C(3,4,K)/S(1,K)
C6=C(6,1,K)
S1K=S(1,K)
41  CALL COMPLEX(C3,C6)
FT3=FT3
FTA=FTA
CALL COMPLEX(C3+DC3,C6)
FT33=(FT3-FT30)/DC3
FT63=(FT6-FT60)/DC3
CALL COMPLEX(C3,C6+DC6)
FT36=(FT3-FT30)/DC3
FTA6=(FTA-FTA0)/DC3
DELTA=FT33+FT66-FT36+FT63
IF(ABS(DELTA),LE,1,9E-5)GO TO 45
DELC3=(FT60+FT36-FT30+FTA6)/DELTA
DELC6=(FT30+FT63-FT60+FT33)/DELTA
ITER=ITER+1
IF(ITER,GT,50)GO TO 47
IF(ABS(DELC3/C3),LE,9,991,AND,ABS(DELC6/C6),LE,9,991)GO TO 45
C3=C3+DELC3
C6=C6+DELC6
IF(C3,LE,0,00000001)C3=(C3-DELC3)/2.
IF(C6,LE,0,00000001)C6=(C6-DELC6)/2.
GO TO 41
45  CAOSS(K)=C6H
C4AOSS(K)=C8R
CC6(K)=C*
C(3,4,K)=C3+S1K
C(6,4,K)=C6+S1K

```



```

      C6HATS(K)=C6HAT
      C(3,1,K)=C3
      RETURN
47  WRITE(4,1012)ITER,C3,CA,FT3,FT6,DELC3,DELC6
1012 FORMAT(1X,F010.00 NOT CONVERGE ON C3*,15,6(1X,1Y,E12.5))
      IERROR=1
      RETURN
C
C   ION EXCHANGE BETWEEN SODIUM AND CALCIUM BY
C   MASS ACTION. NO SURFACTANT COMPLEX.
C   USING NEWTON RAPHSON ITERATION METHOD TO FIND THE CORRECT
C   C4J VALUE
48  N=C(6,4,K)+C6ADSS(K)+C6HATS(K)
48A C4=C(6,J,K)
      NF=1
49  C9=C(3,J,K)+C(5,J,K)-CA
      R96=XK96+C9**2/C6
      C6R=N.5*(2.0+R96-SQRT(4.0+R96+R96**2))
      RHAT=XKHAT+C9**2/CA
      C6HAT=N.5*(2.0+C3R+RHAT-SQRT(4.0+C3R+RHAT+RHAT**2))
      C(6,1,K)=CA*C(1,1,K)/C(1,J,K)
      C(6,2,K)=CA*C(1,2,K)/C(1,J,K)
      C(6,3,K)=C6*C(1,3,K)/C(1,J,K)
      C(6,4,K)=C(6,1,K)*S(1,K)+C(6,2,K)*S(2,K)+C(6,3,K)*S(3,K)
      F6=C(6,4,K)+C6R+C6HAT-B6
      IF(NF.EQ.2) GO TO 50
      CA=C6+NC3
      F5=F6
      NF=2
      GO TO 49
50  FP=(F6-F5)/DC3
      C(6,J,K)=C6-F6/FP
      ITER=ITER+1
      IF(ITER.GT.40) GO TO 47
      IF(C(6,J,K).LT.0.0)C(6,J,K)=C6/2.
      IF(ABS((C(6,J,K)-C6)/C6).GT.0.0001) GO TO 48A
      C6ADSS(K)=C6R
      C6HATS(K)=C6HAT
      RETURN
      END

```

```

      SUBROUTINE COMPLEX(X3,X6)
C-----
C   CALCULATE THE CONCENTRATIONS OF COMPLEX,SODIUM,CALCIUM,SURFACTANT
C   IN AQUEOUS PHASE AND IN ROCK PHASE
C-----
C
      COMMON/HH8/XKC,XK96,XKA6,XKHAT,RV
      COMMON/HH9/C5,C3B,C6R,C6HAT,RB3,R6,FT3,FT6,S1K,C8,C8R
      CR=XKC*X6*X3
      C9=X3+C5-X6-C8
      R96=XK96+C9**2/X6
      R86=XKA6+C9**2/X6
      RETA=R96+SQRT(R96+R86)
      C9H=N.5*(-RETA+SQRT(RETA**2+4.0+R96))
      C6B=C9R**2/R96
      C6R=RV-C6B-C96
      RHAT=XKHAT+C9**2/X6
      C6HAT=N.5*(2.0+C3B+RHAT-SQRT(4.0+C3B+RHAT+RHAT**2))
      C9HAT=C3R-C6HAT
      FT3=C3R+X3*S1K+CR+S1K+C8R-RB3
      FT6=X6+S1K+C6R+C6HAT+2.0*(CA+S1K+C8R)-B6
      RETURN
      END

```

```

SUBROUTINE CHEMADN(X3,ADX3,OLDA3D)
C-----
C PARTIALLY REVERSIBLE LANGMUIR-TYPE ADSORPTION FOR SURFACTANT.
C ADSORPTION IS REVERSIBLE WITH SALINITY
C BUT IRREVERSIBLE WITH SURFACTANT CONCENTRATION.
C
C EXPLANATION OF VARIABLES
C C3PH : SURFACTANT CONCENTRATION IN MOST SURFACTANT RICH PHASE
C A3D : SALINITY DEPENDENT PARAMETER FOR ADSORPTION CALCULATION
C B3D : SALINITY INDEPENDENT PARAMETER FOR ADSORPTION CALCULATION
C
C X3 : IN = C(3,4,K) : TOTAL SURFACTANT CONCENTRATION INCLUDING
C : OUT = C(3,5,K) : TOTAL SURFACTANT CONCENTRATION ONLY IN
C : MOBILE PHASES
C ADX3 : IN = OLD TIMELEVEL SURFACTANT ADSORPTION
C : OUT = NEW TIMELEVEL SURFACTANT ADSORPTION
C OLDA3D : IN = PARAMETER A3D AT OLD TIMELEVEL
C : OUT = PARAMETER A3D AT NEW TIMELEVEL
C-----
C
COMMON/CHEMAD/C3PH,A3D,B3D
C3ADS=A3D*C3PH/(1.+B3D*C3PH)
C3ADSI=A3D*ADX3/OLDA3D
IF(C3ADSI,GE,ADX3)C3ADSI=ADX3
DC3ADS=C3ADS-C3ADSI
IF(DC3ADS,LE,.00001)GO TO 2
IF(C3ADS,GE,X3) GO TO 1
X3=X3-C3ADS
ADX3=C3ADS
GO TO 3
1 ADX3=X3
X3=0.0
GO TO 3
2 X3=X3-C3ADSI
ADX3=C3ADSI
3 OLDA3D=A3D
RETURN
END

```

```

SUBROUTINE POLYADN(X4,ADY4)
C-----
C IRREVERSIBLE LANGMUIR-TYPE ADSORPTION FOR POLYMER
C-----
C
COMMON/POLYAD/C4PH,A4D,B4D
C
C C4ADS=A4D*C4PH/(1.+B4D*C4PH)
DC4ADS=C4ADS-ADX4
IF(DC4ADS,LE,.00001)RETURN
IF(DC4ADS,GE,X4)GO TO 1
X4=X4-DC4ADS
ADX4=C4ADS
RETURN
1 ADX4=X4+ADX4
X4=0.0
RETURN
END

```

```

SUBROUTINE MATBAL
C-----
C THIS SUBPROGRAM CALCULATE MATERIAL BALANCE ERROR,
C GRID CONSTRUCTION IS BLOCK CENTERED GRID,
C
C ZI = INJECTED AMOUNT
C ZE = INITIALLY EXISTED AMOUNT
C PR = PRODUCED AMOUNT
C ADS = ADSORBED AMOUNT
C CMOB = AMOUNT THAT EXIST IN MOBILE PHASE
C-----
C
C DIMENSION ADS(7),PR(7),CMOB(7),E(7),RE(7)
C
COMMON/NO/ICT,ICT1,ICT2,XICT,NCOMP
COMMON/PROP/IN/ER,P(7),PA,ZI(7),ZE(7),S2
COMMON/SOL/C(7,4,42),S(3,42),FF(3,42),NPHASE(42)
COMMON/SYSTEM/IT,ABPERM,PHI,EPHI3,EPHI4,DISP(4)
COMMON/ADSORP/C3ADSS(40),C6ADSS(40),C6ADSS(40),C6HATS(40)
COMMON/CA/C6ADSS(40),CC8(42)
C
DO 10 I=1,7
10 CMOB(I)=ADS(I)*S,0
C6HAT=0,0
DO 20 K=1,ICT
X=1,-C3ADSS(K)
ADS(3)=ADS(3)+C3ADSS(K)*XICT+C6ADSS(K)*XICT
ADS(4)=ADS(4)+C6ADSS(K)*XICT
ADS(6)=ADS(6)+C6ADSS(K)*XICT+C6ADSS(K)*XICT*2.
C6HAT=C6HAT+C6HATS(K)*XICT
CMOB(1)=CMOB(1)+C(1,4,K)*XICT*X
CMOB(2)=CMOB(2)+C(2,4,K)*XICT*X
CMOB(3)=CMOB(3)+C(3,4,K)*XICT*X+CC8(K)*S(1,K)*XICT
CMOB(4)=CMOB(4)+C(4,4,K)*XICT
CMOB(5)=CMOB(5)+C(5,4,K)*XICT
CMOB(6)=CMOB(6)+C(6,4,K)*XICT+CC8(K)*S(1,K)*XICT*2.
20 CMOB(7)=CMOB(7)+C(7,4,K)*XICT*X
CMOB(3)=CMOB(3)+EPHI3
CMOB(4)=CMOB(4)+EPHI4
ADS(3)=ADS(3)+EPHI3
ADS(4)=ADS(4)+EPHI4
ADS(6)=ADS(6)+C6HAT
DO 30 J=1,NCOMP
30 PR(J)=P(J)
PR(3)=PR(3)+P8
PR(6)=PR(6)+P8*2.
DO 50 I=1,NCOMP
IF((ZI(I)+ZE(I)).LE.0,00001) GO TO 51
E(I)=PR(I)+ADS(I)+CMOB(I)-ZI(I)-ZE(I)
RE(I)=E(I)/(ZI(I)+ZE(I))
GO TO 50
51 E(I)=0,0
50 CONTINUE
PRINT 100
DO 60 I=1,NCOMP
60 PRINT 110,I,ZE(I),ZI(I),PR(I),ADS(I),CMOB(I),E(I),RE(I)
100 FORMAT(1H1,///5X,MATERIAL BALANCE BASED ON BLOCK CENTERED GRID///
1 //5X,9HCOMPONENT,6X,9HINITIALLY,6X,5HTOTAL,10X,5HTOTAL,10X,
1 5HADSORBED,6X,6HMOBILE,6X,5HABSOLUTE,6X,5HRELATIVE/
1 6X,2HIN,9X,7HEXISTED,7X,6HINJECTED,7X,6HPRODUCED,6X,6HAMOUNT,
1 10X,5HPHASE,10X,5HERROR,10X,5HERROR)

```

```

110 FORMAT(/6X,11,6X,7E15,5)
RETURN
END

```

SEMI-DISCRETE WITH RK12
NO ADSORPTION
DATA FILE FOR JAPEX2
SEMI-DISCRETE METHOD

RK1 IS USED

HIRASAKI'S MODEL IS USED IF THREE PHASES APPEAR

INPUT VALUES

NSLUG= 2

ISOLV= 1	ISEM= 2	METH= 3	MITER= 0	IPERM= 1	ID= 40
VI= .4100	FFDV= .0000	NCOMP= 6	ICT= 40	ICTL= 1	ICTU= 40
UT= 0.0000	APPERM= 1.0000	PHI= .2000	EPH13= 1.0000	EPH14= 1.0000	DISP= 0.0000
CSI1= .0000	C611= 0.0000	S1= .6300	S2= .3700	S1H= .3700	S2H= .3700
G11= 6.2850	G12= -7.0580	G13= .1100	G21= 6.2850	G22= -7.0580	G23= .1100
T11= .3700	T12= 2.8700	T21= .3700	T22= .9000	XIFTW= 1.3000	
ALPHA1= 0.0000	ALPHA2= 0.0000	ALPHA3= 50.0000	ALPHA4= 0.0000	ALPHA5= 0.0000	
VIS1= 1.0000	VIS2= 5.0000	AP1= 100.0000	AP2= 100.00	AP3= 10000.00	SSLOPE= 0.0000
GAMHF= 13.6000	P0H= 1.0000	CSE1= .0100	RKMAX= 0.0000	BRK= 0.0000	
P1H= .0500	P2H= 1.0000	E1= 1.5000	E2= 1.5000	P1RC= 1.0000	P2RC= 1.0000
C3MAX3= .3000	C3MAX1= .1000	C3MAX2= .3000			
C2PLC= 0.0000	C2PRC= 1.0000	CSEL= .0000	CSEU= 1.2000		
QV= 0.0000	RCSE= 1.0000	AD31= 0.0000	AD32= 0.0000	B3D= 100.0000	
A4D= 0.0000	B4D= 100.0000	XK9A= 0.0000	XK8A= 0.0000	XKC= 0.0000	XKHAT= 0.0000

COMPOSITION OF INJECTED SLUG

SLUG	CUMUL.	NO	INJ. VOL.	C1	C2	C3	C4	C5	C6	C7
1	.100	.90000	0.00000	.10000	.10000	.70000	1.00000			
2	.400	1.00000	0.00000	0.00000	.10000	.50000	1.00000			

Example Output

CALCULATED A PARAMETERS FOR BINODAL CURVE

A11= -.6559 A12= .6853 A21= .7347 A22= -.6853

INITIAL CONDITIONS

WATER SATURATION	:	.6300
OIL SATURATION	:	.3700
WATER PHASE RELATIVE PERM.	:	.2500
OIL PHASE RELATIVE PERM.	:	0.0000
WATER VISCOSITY	:	1.0000
OIL VISCOSITY	:	5.0000

PRESSURE DROP RECORDED BETWEEN 1 AND 00

RESIDUAL WATER SATURATION	:	.3700
RESIDUAL OIL SATURATION	:	.3700
END POINT REL. PERM. FOR WATER	:	.0500
END POINT REL. PERM. FOR OIL	:	1.0000
WATER FRACTIONAL FLOW	:	1.0000
OIL FRACTIONAL FLOW	:	0.0000

PCT = .800

VP	P1	P2	P3	P4	P5	P6	P7	ER	PR	TOT.MOR.	PRESSURE	IEVA
.132	.2627	.0691	.0000	.2101	.0000	.0000	.0000	.1869	.0000	.0546	1.2713	284
PHASE CONCENTRATION OF COMPONENTS												
	C1	C2	C3	C4	C5	C6	C7	PHASE CUT				
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6400				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3600				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.341	.2685	.0724	.0000	.2149	.0000	.0000	.0000	.1956	.0000	.0545	1.2947	288
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6422				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3578				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.351	.2747	.0759	.0000	.2198	.0000	.0000	.0000	.2050	.0000	.0544	1.3045	294
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6427				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3573				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.361	.2814	.0796	.0000	.2252	.0000	.0000	.0000	.2151	.0000	.0544	1.3149	301
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6431				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3569				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.371	.2882	.0833	.0000	.2306	.0000	.0000	.0000	.2251	.0000	.0544	1.3431	306
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6457				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3543				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.381	.2943	.0866	.0000	.2350	.0000	.0000	.0000	.2342	.0000	.0543	1.3519	313
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6478				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3522				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				
.391	.3006	.0901	.0000	.2400	.0000	.0000	.0000	.2434	.0000	.0542	1.3770	317
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.6471				
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3529				
3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	.0000				
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0000				

PROFILE AT 1000 P.V. INJECTED

XU	.175 (NPHASE=2)				.154 (NPHASE=2)				.125 (NPHASE=2)			
CSE,CSEP	.5476 .5476				.5413 .5413				.5045 .5045			
MUR, DELP	.4284 2.5445				.4281 2.4941				.4203 2.4573			
PHASE	1	2	3	TOTAL	1	2	3	TOTAL	1	2	3	TOTAL
SATURATION	0.0000	.1442	.8518		0.0000	.1470	.8530		0.0000	.1436	.8564	
C1	0.0000	0.0000	.9971	.8493	0.0000	0.0000	.9987	.8518	0.0000	0.0000	.9998	.8541
C2	0.0000	1.0000	.0000	.1482	0.0000	1.0000	.0000	.1470	0.0000	1.0000	.0000	.1438
C3	0.0000	0.0000	.0029	.0025	0.0000	0.0000	.0013	.0011	0.0000	0.0000	.0002	.0002
C4	0.0000	0.0000	.0000	.0000	0.0000	0.0000	.0000	.0000	0.0000	0.0000	.0000	.0000
C5	0.0000	0.0000	.5462	.4311	0.0000	0.0000	.5428	.4288	0.0000	0.0000	.5011	.4281
C6	0.0000	0.0000	.9988	.8508	0.0000	0.0000	.9996	.8526	0.0000	0.0000	.9999	.8543
RES. SAT.	0.0000	.1442	.3700		0.0000	.1470	.3700		0.0000	.1436	.3700	
NORM. SAT.	0.0000	0.0000	1.0000		0.0000	0.0000	1.0000		0.0000	0.0000	1.0000	
REL. PERM	0.0000	0.0000	.6195		0.0000	0.0000	.6225		0.0000	0.0000	.6309	
VISCOSITY	0.0000	5.0000	30.9800		0.0000	5.0000	30.9999		0.0000	5.0000	31.0015	
FRAC. FLOW	0.0000	0.0000	1.0000		0.0000	0.0000	1.0000		0.0000	0.0000	1.0000	
IFT (DYNE/CM)	MM	MO	NO		MM	MO	NO		MM	MO	NO	
	0.00000	.16665	0.00000		0.00000	.16775	0.00000		0.00000	.16830	0.00000	
ADSORPTION	C3	C4	C6	C6HATS	C3	C4	C6	C6HATS	C3	C4	C6	C6HATS
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
COMPLEX	CR	CRADSS			CR	CRADSS			CR	CRADSS		
	0.0000	0.0000			0.0000	0.0000			0.0000	0.0000		

XU	.075 (NPHASE=2)				.050 (NPHASE=2)				.025 (NPHASE=2)			
CSE,CSEP	.5401 .5401				.5000 .5000				.5000 .5000			
MUR, DELP	.4285 2.4354				.4287 2.4127				.4342 1.6536			
PHASE	1	2	3	TOTAL	1	2	3	TOTAL	1	2	3	TOTAL
SATURATION	.8584	.1416	0.0000		.8607	.1393	0.0000		.8625	.1375	0.0000	
C1	1.0000	0.0000	0.0000	.8584	1.0000	0.0000	0.0000	.8607	1.0000	0.0000	0.0000	.9756
C2	0.0000	1.0000	0.0000	.1416	0.0000	1.0000	0.0000	.1393	0.0000	1.0000	0.0000	.0244
C3	0.0000	0.0000	0.0000	.0000	0.0000	0.0000	0.0000	.0000	0.0000	0.0000	0.0000	.0000
C4	.1400	0.0000	0.0000	.0858	.1000	0.0000	0.0000	.0861	.1000	0.0000	0.0000	.0976
C5	.5001	0.0000	0.0000	.4293	.5000	0.0000	0.0000	.4304	.5000	0.0000	0.0000	.4878
C6	1.0000	0.0000	0.0000	.8584	1.0000	0.0000	0.0000	.8607	1.0000	0.0000	0.0000	.9756
RES. SAT.	.3700	.1416	0.0000		.3700	.1393	0.0000		.3700	.1375	0.0000	
NORM. SAT.	1.0000	0.0000	0.0000		1.0000	0.0000	0.0000		1.0000	0.0000	0.0000	
REL. PERM	.6365	0.0000	0.0000		.6424	0.0000	0.0000		.6469	0.0000	0.0000	
VISCOSITY	31.0000	5.0000	0.0000		31.0000	5.0000	0.0000		31.0000	5.0000	0.0000	
FRAC. FLOW	1.0000	0.0000	0.0000		1.0000	0.0000	0.0000		1.0000	0.0000	0.0000	
IFT (DYNE/CM)	MM	MO	NO		MM	MO	NO		MM	MO	NO	
	0.00000	0.00000	19.95262		0.00000	0.00000	19.95262		0.00000	0.00000	19.95262	
ADSORPTION	C3	C4	C6	C6HATS	C3	C4	C6	C6HATS	C3	C4	C6	C6HATS
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
COMPLEX	CR	CRADSS			CR	CRADSS			CR	CRADSS		
	0.0000	0.0000			0.0000	0.0000			0.0000	0.0000		

MATERIAL BALANCE BASED ON BLOCK CENTERED GRID

COMPONENT NO	INITIALLY EXISTED	TOTAL INJECTED	TOTAL PRODUCED	ABSORBED AMOUNT	MOBILE PHASE	ABSOLUTE ERROR	RELATIVE ERROR
1	.63000E+00	.30000E+00	.30667E+00	0.	.71333E+00	-.24869E-13	-.24381E-13
2	.37000E+00	0.	.93335E-01	0.	.27667E+00	-.49738E-13	-.13443E-12
3	0.	.10000E-01	.10523E-00	0.	.10000E-01	.13878E-14	.13978E-12
4	0.	.40000E-01	.10525E-00	0.	.40000E-01	-.10984E-14	-.49960E-13
5	.50000E+00	.22000E+00	.24533E+00	0.	.47867E+00	-.42988E-12	-.59375E-12
6	0.	.40000E+00	.10525E-07	0.	.40000E+00	-.35527E-14	-.88818E-14

NUMBER OF REJECTION = 18

NUMBER OF FUNCTION EVALUATION = 323

REFERENCES

1. Reed, R. L. and Healy, R. N., "Some physicochemical aspects of microemulsion flooding: A review," in Improved Oil Recovery by surfactant and polymer flooding, Shah, D.D. and Schechter, R.S., ed., Academic Press, New York, New York, 1977, 383-437.
2. Nelson, R.C. and Pope, G.A., "Phase relationship in chemical flooding," Soc. Pet. Eng. J. (Oct. 1978), 325-338.
3. Uren, L.C. and Fahmy, E.H., "Factors influencing the recovery of petroleum from unconsolidated sands by waterflooding," Trans. AIME, 1927.
4. van Poolen, H.K., Fundamentals of Enhanced Oil Recovery, Penn Well Books, Tulsa, Oklahoma, 1980.
5. Pope, G.A. and Nelson, R.C., "A chemical flooding compositional simulator," Soc. Pet. Eng. J. (October, 1978), 339-354.
6. Thomas, C.P., Winter, W.K. and Fleming, P.D., III, "Application of a general multiphase, multicomponent chemical flooding model to ternary two-phase surfactant systems," SPE 6727, presented at 52nd Annual Fall Technical Conference of SPE, Denver, Colorado, Oct. 9-12, 1977.
7. Larson, R.G., "The influence of phase behavior on surfactant flooding," Soc. Pet. Eng. J. (Dec. 1979), 411-422.
8. Pope, G.A., Wang, Ben and Tsaor, K., "A sensitivity study of micellar/polymer flooding," Soc. Pet. Eng. J. (Dec. 1979), 357-368.
9. Hirasaki, G.J., van Domselaar, H.R., Nelson, R.C., "Evaluation of the salinity gradient concept in surfactant flooding," SPE 8825, presented at the First Joint SPE/DOE Symposium on Enhanced Oil Recovery, Tulsa, Oklahoma, April 20-23, 1980.
10. Todd, M.R., Dietrich, J.K. Goldberg, A. and Larson, R.G., "Numerical simulation of competing chemical flood designs," SPE 7077, presented at the Fifth Symposium on Improved Methods for Oil Recovery of SPE, Tulsa, Oklahoma, April 16-19, 1978.
11. Todd, M.R., and Chase, C.A., "A numerical simulation for predicting chemical flood performance," SPE 7689, presented at the Fifth SPE Symposium on Reservoir Simulation, Denver, Colorado, Feb. 1-2, 1979.

12. Pang, H.W. and Caudle, B. H., "Modeling of a micellar-polymer process," SPE9009, presented at the SPE Fifth International Symposium on Oilfield and geothermal chemistry, Stanford, California, May 28-30, 1980.
13. Pope, G.A., "Mobility control and scaleup for chemical flooding," First Annual Report to DOE, the University of Texas at Austin, Austin, Texas, 1980.
14. Pope, G.A., Hong, C.H., Sepehrnoori, K. and Lake, L.W., "Two dimensional simulation of chemical flooding," SPE 9939, presented at SPE California Regional Meeting, Bakersfield, California, March 25-27, 1981.
15. Chase, C.A., "Variational simulation with numerical decoupling and local mesh refinement," SPE 7680, presented at the Fifth SPE Symposium on Reservoir Simulation, Denver, Colorado, Feb. 1-2, 1979.
16. Kazemi, H. and MacMillan, D.J., "A numerical simulation comparison of five spot vs. line drive in micellar-polymer SPE9427, presented at the 55th Annual Fall Technical Conference of SPE, Dallas, Texas, Sept. 21-24, 1980.
17. Peaceman, D.W., Fundamentals of Numerical Reservoir Simulation, Elsevier North-Holland Inc., New York, New York, 1977.
18. Lantz, R.B., "Qualitative evaluation of numerical diffusion (truncation error)," Soc. Pet. Eng. J. (Sept. 1971), 315-320.
19. Perkins, T.K. and Johnston, O.C., "A review of diffusion and dispersion in porous media," Soc. Pet. Eng. J. (March 1963), 70-84.
20. Collins, R.E., Flow of Fluids Through Porous Materials, the Petroleum Publishing Company, Tulsa, Oklahoma, 1976.
21. Price, H.S., Cavendish, J.C., and Varga, R.S., "Numerical methods of higher-order accuracy for diffusion-convection equations," Soc. Pet. Eng. J. (Sept. 1968), 293-303.
22. Spivak, A., Price, H.S. and Settari, A., "Solution of the equations for multidimensional, two phase, immiscible flow by variational methods," Soc. Pet. Eng. J. (Feb. 1977), 27-41.
23. Settari, A., Price, H.S. and Dupont, T., "Development and application of variational methods for simulation of miscible displacement in porous media," Soc. Pet. Eng. J. (June, 1977), 228-246.

24. Sepehrnoori, K., Carey, G.F. and Knapp, R., "Convection-diffusion computations," Proceedings of the Third International Conference in Australia on Finite Element Methods, July 1979, The University of New Wales.
25. Jensen, O.K., "Numerical modeling with a moving coordinate system: application to flow through porous media," Ph.D. dissertation, University of Washington, Seattle, Washington, 1980.
26. Peaceman, D.W. and Rachford, H.H., Jr., "Numerical calculation of multidimensional miscible displacement," Soc. Pet. Eng. J. (Dec. 1962), 327-339; Trans., AIME, Vol. 225.
27. Price, H.S., Varga, R.S. and Warren, J.E., "Application of oscillation matrices to diffusion-convection equations," J. of Math. and Phys. (Sept. 1966), Vol. 45, 301-311.
28. Chaudhari, N.M., "An improved numerical technique for solving multidimensional miscible displacement equations," Soc. Pet. Eng. J. (Sept. 1971), 277-284; Trans., AIME, Vol. 251.
29. Todd, M.R., O'Dell, P.M. and Hirasaki, G.J., "Methods for increased accuracy in numerical reservoir simulators," Soc. Pet. Eng. J. (Dec. 1972), 515-530; Trans. AIME, Vol. 253.
30. Laumbach, D.D., "A high-accuracy finite-difference technique for treating the convection-diffusion equation," Soc. Pet. Eng. J. (Dec. 1975), 517-531.
31. Yokoyama, Y., "The effects of capillary pressure on displacements in stratified porous media," M.S. thesis, The University of Texas at Austin, Austin, Texas, 1981.
32. Garder, A.O., Jr., Peaceman, D.W., Pozzi, A.I., Jr., "Numerical calculation of multidimensional miscible displacement by the method of characteristics," Soc. Pet. Eng. J. (March 1964), 26-36; Trans. AIME, Vol. 231.
33. Nolen, J.S., "Numerical simulation of compositional phenomena in petroleum reservoirs," SPE 4274, presented at the Third Symposium on Numerical Simulation of Reservoir Performance of SPE, Houston, Texas, Jan. 11-12, 1973.
34. Ben-Omran, A.M. and Green, D.W., "A two-dimensional, two phase compositional model which uses a moving point method," SPE 7415, presented at the 53rd Annual Fall Technical Conference of SPE, Houston, Texas, Oct. 1-3, 1978.

35. Naiki, M., "Numerical simulation of polymer flooding including the effects of salinity," Ph.D. dissertation, The University of Texas at Austin, Austin, Texas, 1979.
36. Larson, R.G., "Controlling numerical dispersion by variably timed flux updating in two dimensions," SPE9374, presented at the 55th Annual Fall Technical Conference of SPE, Dallas, Texas, Sept. 21-24, 1980.
37. Engelsen, S., Lake, L.W., Ohno, T., Pope, G.A., Sepehrnoori, K. and Wang, B., "A description of a generalized micellar/polymer physical property package," Center for Enhanced Oil and Gas Recovery Research, University of Texas at Austin, Austin, Texas, 1981.
38. Wang, Ben, "A sensitivity study of micellar polymer flooding," M.S. thesis, the University of Texas at Austin, Austin, Texas, 1977.
39. Lin, E., "A study of micellar/polymer flooding using a compositional simulator," Ph.D. dissertation, the University of Texas at Austin, Austin, Texas, 1981.
40. Coats, K.H., "A highly implicit steamflood model," Soc. Pet. Eng. J. (Oct. 1978), 369-383.
41. Grabowski, J.W., Vinsome, P.K., Lin, R.C., Behie, A., and Rubin, B., "A fully implicit general purpose finite difference thermal model for in situ combustion and steam," SPE8396, presented at the 54th Annual Fall Technical Conference of SPE, Las Vegas, Nevada, Sept. 23-26, 1979.
42. Sincovec, R.F., "Numerical reservoir simulation using an ordinary differential equations integrator," Soc. Pet. Eng. J. (June 1975), 255-264.
43. Hindmarsh, A.C., "GEARB: solution of ordinary differential equations having banded Jacobian," UCID-30059, Lawrence Livermore Laboratory (May 1973).
44. Jensen, O.K., "An automatic timestep selection scheme for reservoir simulation," SPE9373, presented at the 55th Annual Fall Technical Conference of SPE, Dallas, Texas, Sept. 21-24, 1980.
45. Sepehrnoori, K. and Carey, G.F., "Numerical integration of semi-discrete evolution systems," to appear in the Journal of Computer Methods in Applied Mechanics and Engineering.

46. Hill, H. J. and Lake, L.W., "Cation exchange in chemical flooding: part 3--experimental," Soc. Pet. Eng. J. (Dec. 1978), 445-456.
47. Engelsen, S., "Micellar/polymer flooding simulation--improvement in modeling and matching of core floods," M.S. thesis, the University of Texas at Austin, Austin, Texas, 1981.
48. Glover, C.L., Puerto, M.C., Maerker, J.M., and Sandvik, E.L., "Surfactant phase behavior and retention in porous media," Soc. Pet. Eng. J. (June 1979), 183-193.
49. Nelson, R.C., "Further studies on phase relationships in chemical flooding," presented at the Third Conference on Surface and colloid science, Stockholm, Sweden, August 20-25, 1979.
50. Nelson, R.C., "The salinity requirement diagram--a useful tool in chemical flooding research and development," SPE8824, presented at the First Joint SPE/DOE Symposium on Enhanced Oil Recovery, Tulsa, Oklahoma, April 20-23, 1980.
51. Treybal, R.E., Liquid Extraction, 2nd ed., McGraw Hill Book Co., Inc., New York City, 1963, 37.
52. Langmuir, I., J. Am. Chem. Soc., 38, 2267 (1916).
53. Stegemeir, G.L., "Mechanism of entrapment and mobilization of oil in porous media," in Improved Oil Recovery by Surfactant and Polymer Flooding, Shah, D.O. and Schechter, R.S., ed., Academic Press, New York, New York, 1979, 384-437.
54. Taber, J.J., "Dynamic and static forces required to remove a discontinuous oil phase from porous media containing both oil and water," Soc. Pet. Eng. J. (March 1969), 3-12.
55. Gupta, S.P. and Trushenski, S.P., "Micellar flooding--compositional effects on oil displacement," Soc. Pet. Eng. J. (April 1979), 116-128.
56. Fehlberg, E., "Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control," NASA TR R-287, October, 1968.
57. Fehlberg, E., "Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems," NASA TR R-315, July, 1969.
58. Gear, C.W., Numerical Initial-Value Problems in Ordinary Differential Equations, Prentice Hall, Englewood Cliffs, N.J., 1971.

59. Algorithm DGEAR, IMSL Library Vol. 1, Edition 7, 1979.
60. Atkinson, K.E., An Introduction to Numerical Analysis, John Wiley & Sons, Inc., 1978.
61. Sepehrnoori, K., "Runge-Kutta formulas with extended regions of stability," TICOM Report 78-10, The University of Texas at Austin, Austin, Texas, May 1978.
62. Shampine, L., Watts, H., "Practical solution of ordinary differential equations by Runge-Kutta methods," Sandia Labs Tech. Rep. SAND76-0585, Albuquerque, New Mexico, 1976.
63. Lake, L.W., Pope, G.A., Carey, G.F., Sepehrnoori, K., "Isothermal, multiphase, multicomponent fluid-flow in permeable media Part I: Description and mathematical formulation," Center for Enhanced Oil and Gas Recovery Research, The University of Texas at Austin, Austin, Texas,
64. McKenzie, R.E. and Sepehrnoori, K., "An implementation of an eighth order Runge-Kutta-Fehlberg method", IASOM TR78-6, The University of Texas at Austin, Austin, Texas, December 1978.
65. Fil, A., Personal Communication, 1981.
66. Hirasaki, G.J., "Ion exchange with clays in the presence of surfactant", SPE9279, presented at the 55th Annual Fall Technical Conference of SPE, Dallas, Texas, Sept. 21-24, 1980.
67. Fleming, P.D., III, Thomas, C.P., Winter, W.K., "Formulation of a general multiphase, multicomponent chemical flood model", Soc. Pet. Eng. J. (Feb. 1981), 63-76.
68. Kossac, C.A. and Bilhartz, H.L., Jr., "The sensitivity of micellar flooding to reservoir heterogeneity", SPE5808, presented at Improved Oil Recovery Symposium of SPE, Tulsa, Oklahoma, March 22-24, 1976.

VITA

Takamasa Ohno, son of Yoshiharu Ohno and Tomoko Ohno was born in Hiroshima, Japan, on February 23, 1952. In April, 1970, he entered the University of Tokyo. He received the Bachelor of Engineering degree in Petroleum Engineering in 1974. During the following five years, he worked for the Japan Petroleum Exploration Corporation (JAPEX) as a production engineer. In January, 1979, he entered the Graduate School of the University of Texas at Austin. He married Mihoko Imaichi in May, 1977 and has one son Takeshi.

Permanent address: c/o Japan Petroleum Exploration Co., Ltd.
Production Department
No. 6-1, 1-Chome,
Ohtemachi, Chiyoda-ku
Tokyo, Japan

This thesis was typed by Anderson Typing Service.